

# **Software-Defined Networking**

Was? Wo? Weshalb? Wie?

**Stefan Schmid**

TU Berlin & Telekom Innovation Labs (T-Labs)

# Software-Defined Networking

Was? **Wo?** Weshalb? Wie?

Stefan Schmid

Wo wird SDN eingesetzt?

-Labs)

# Software-Defined Networking

Was? Wo? Weshalb? Wie?

Schmid

1. “Wichtigster Durchbruch seit der Erfindung des Internets” [SDN Academy]

abs (T-Labs)

# Software-Defined Networking

Was? Wo? Weshalb? Wie?

Schmid

1. “Wichtigster Durchbruch seit der Erfindung des Internets” [SDN Academy]

abs (T-

2. “Weltweiter SDN Markt für Unternehmen und Cloud Service Providers wird bis 2016 schätzungsweise um \$3.7 Mia wachsen” [IDC Studie]

# Software-Defined Networking

Was? Wo? Weshalb? Wie?

3. “Netzwerke sind wieder cool “

[Cisco CTO Padmasree Warrior]

Schmid

1. “Wichtigster Durchbruch seit der Erfindung des Internets”

[SDN Academy]

abs (T-

2. “Weltweiter SDN Markt für Unternehmen und Cloud Service Providers wird bis 2016 schätzungsweise um \$3.7 Mia wachsen”

[IDC Studie]

# Software-Defined Networking

Was? Wo? Weshalb? Wie?

3. “Netzwerke sind wieder cool “

[Cisco CTO Padmasree Warrior]

1. “  
Er

Thema dieser Herbsttagung: Programmierbarkeit  
von Netzwerken und Infrastruktur.  
Was steckt dahinter?

2. “Weltweiter SDN Markt für Unternehmen und Cloud  
Service Providers wird bis 2016 schätzungsweise um  
\$3.7 Mia wachsen” [IDC Studie]

# Software-Defined Networking

Was? Wo? Weshalb? Wie?

**Stefan Schmid**

TU Berlin & Telekom Innovation Labs (T-Labs)

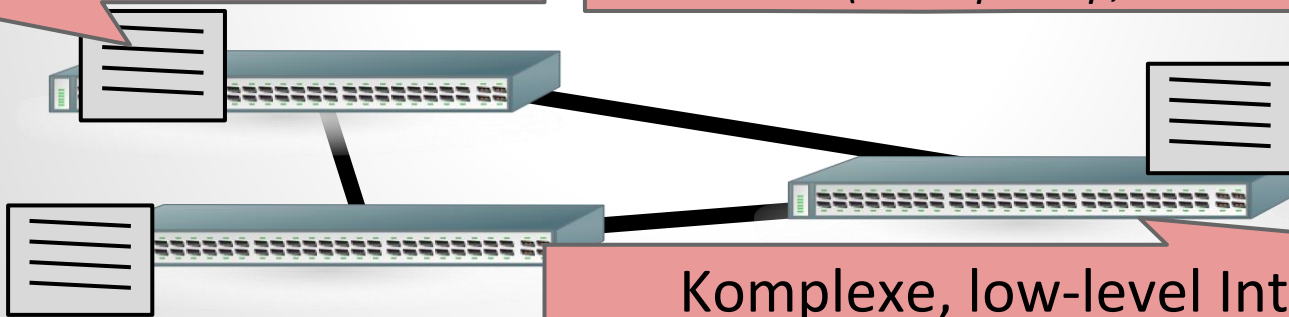
*Einige Folien von: Marco Canini, Dan Levin, Nate Foster  
und Roger Wattenhofer!*

# Netzwerke Heute

Kritische Infrastruktur, aber Management und Operation **mühsam, manuell, fehleranfällig**: unzählige Konfigurationsdateien verteilt über Geräte, **Schnittstellen variieren** zwischen Vendors und sogar zwischen Geräten vom gleichen Vendor, etc.

Manuelle, Geräte-  
zentrische Konfiguration  
(CLI, LANmanager)

“Best Practices” und Troubleshooting  
Tools stammen aus den 90ern  
(z.B. *tcpdump*, *traceroute*)



Komplexe, low-level Interfaces  
(VLANs, *Spanning Tree*, *Routing*)

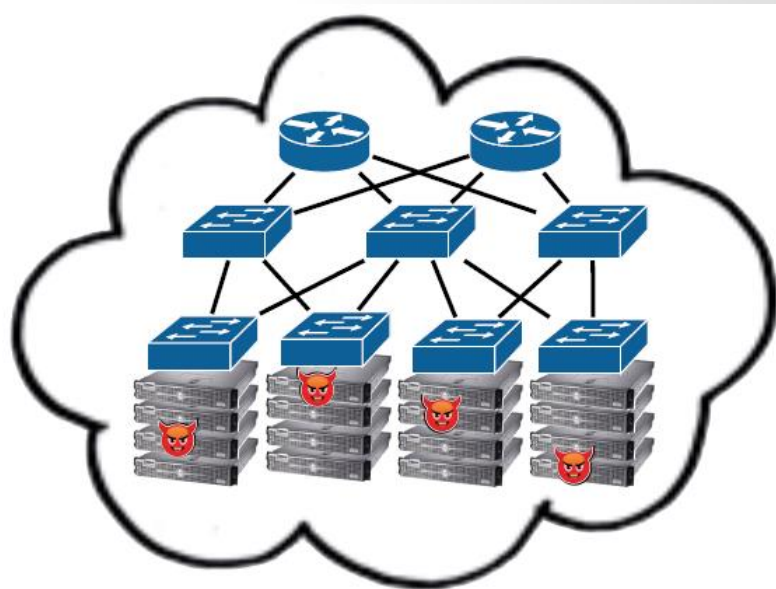


# Weshalb ist das ein Problem?

Ein Beispiel: Immer mehr Firmen outsourcen Infrastruktur in die public, **shared** Cloud (z.B. auch die CIA)

Falsch konfiguriertes Netzwerk:  
Sicherheitsrisiko!

Was wenn der Netzwerkverkehr während  
Unterhaltungsarbeiten (z.B. Firmware  
Update) **nicht isoliert** ist von anderen  
Nutzern / “Tenants”?



# Netzwerkausfälle: Realität

(c) Nate Foster

Selbst technisch versierte Unternehmen wie GitHub oder Amazon haben Mühe, zuverlässige **Netzwerkperformance zu garantieren**.



*We discovered a **misconfiguration** on this pair of switches that caused what's called a **"bridge loop"** in the network.*

*A **network change** was [...] executed incorrectly [...] more "stuck" volumes and added more requests to the **re-mirroring storm***



*Service outage was due to a series of internal network events that **corrupted router data** tables*

*Experienced a network connectivity issue [...] **interrupted the airline's flight departures**, airport processing and reservations systems*



# ZKI Herbsttagung: Aktuelles Thema

Trend: Netzwerke sind mehr und mehr...

- Virtualisiert
- Software-Defined
- Offen

# ZKI Hei

## Trend: Netzwerk

### Vorteile:

- **Entkoppelung** der Anwendung von den Limitierungen der darunterliegenden Infrastruktur
- Flexible **Ressourcenallokation** und Migration
- **Ressourcensharing** und trotzdem **Isolation**

- Virtualisiert
- Software-Defined

### Vorteile:

- **Schnelle Innovation: Programmierbarkeit**
- Ermöglicht flexibleres und **automatisiertes** Netzwerkmanagement

- Offen

### Vorteile:

- **Standardisierte und einheitliche** Schnittstellen um Geräte zu programmieren

# ZKI Hei

## Trend: Netzwerk

### Vorteile:

- **Entkoppelung** der A
- Limiti
- Flexi
- **Manokation** und Migration
- **Ressourcensharing** und trotzdem **Isolation**

Beispiele: OpenStack, NFV, ...

### Vorteile:

- **Schnelle In**
- **Mark**
- **automatisiertes**
- **management**

Beispiele: OpenFlow, Click, ...

### Vorteile:

- **Stand**
- **Schnittstellen**
- **programmieren**

Beispiele: OpenStack, OpenFlow, ...

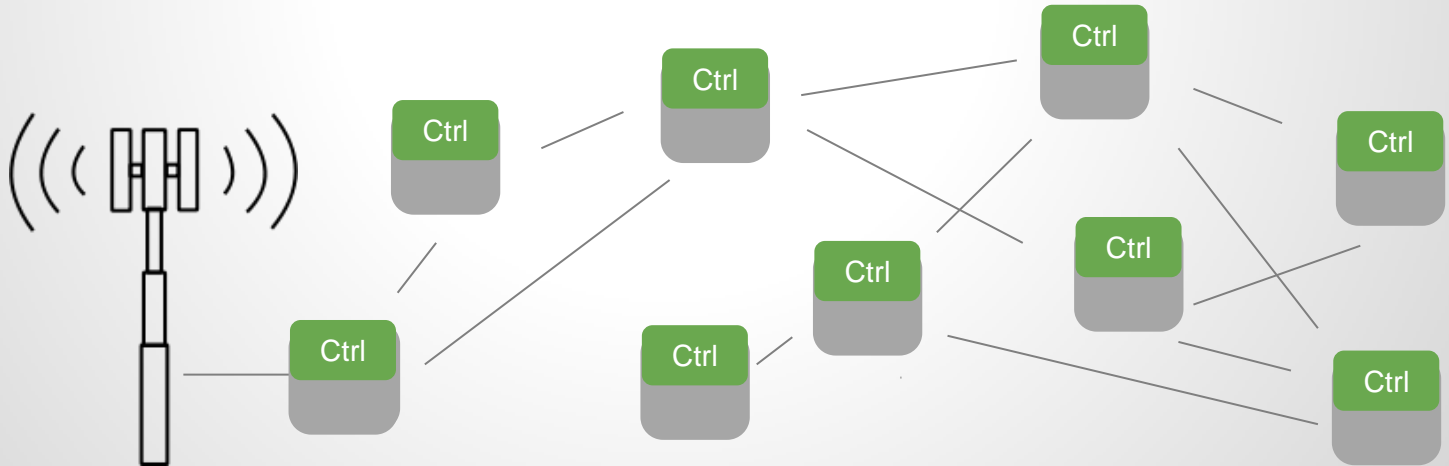
# SDN in a Nutshell

# SDN in a Nutshell

Grundsätzliches

Konzept: **Auslagerung  
der Kontrolle** über

Geräte an einen (logisch)  
zentralisierten Software-  
Kontroller

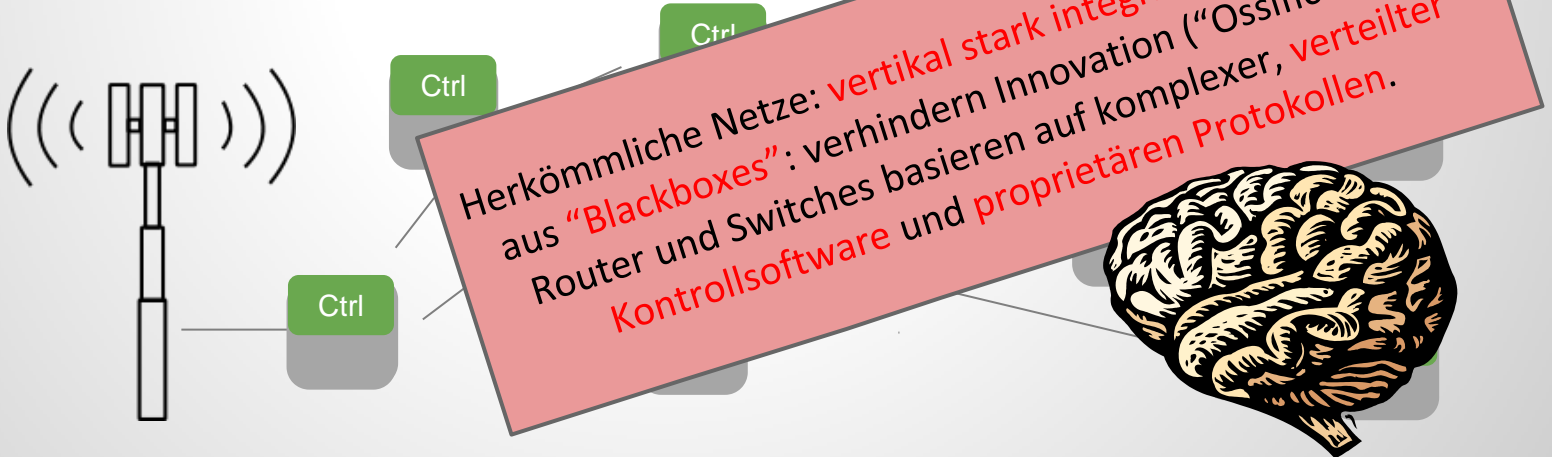


# SDN in a Nutshell

Grundsätzliches

Konzept: **Auslagerung  
der Kontrolle** über

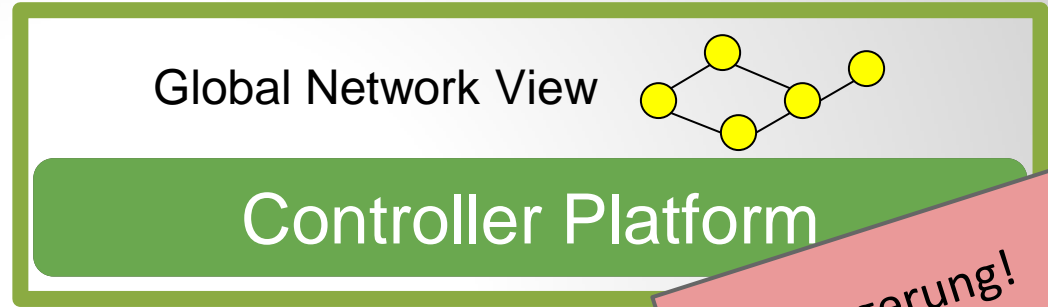
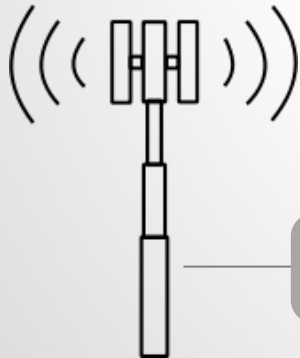
Geräte an einen (logisch)  
zentralisierten Software-  
Kontroller



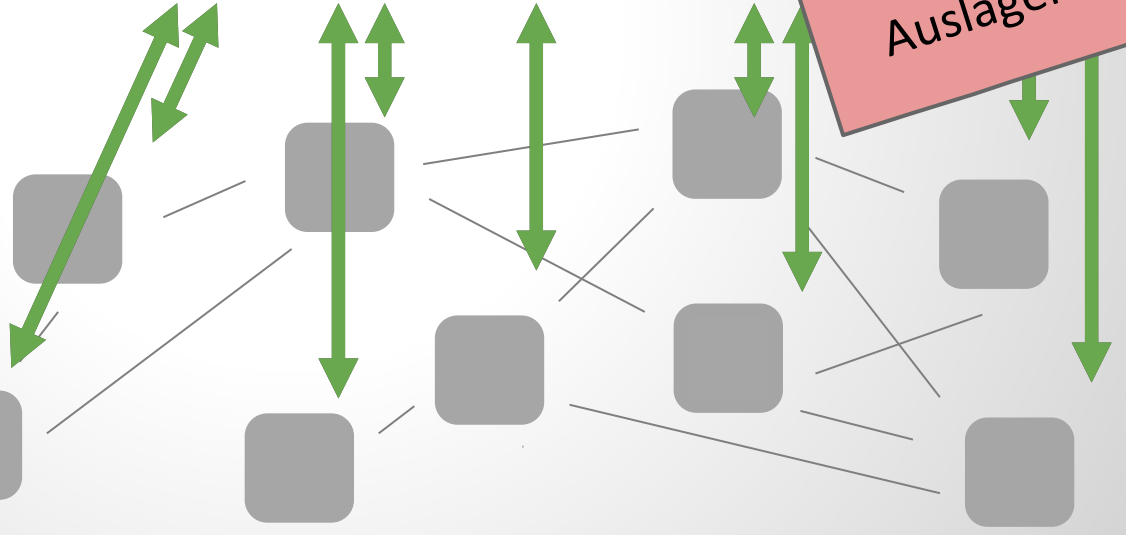


# SDN in a Nutshell

Grundsätzliches  
Konzept: **Auslagerung  
der Kontrolle** über  
Geräte an einen (logisch)  
zentralisierten Software-  
Kontroller

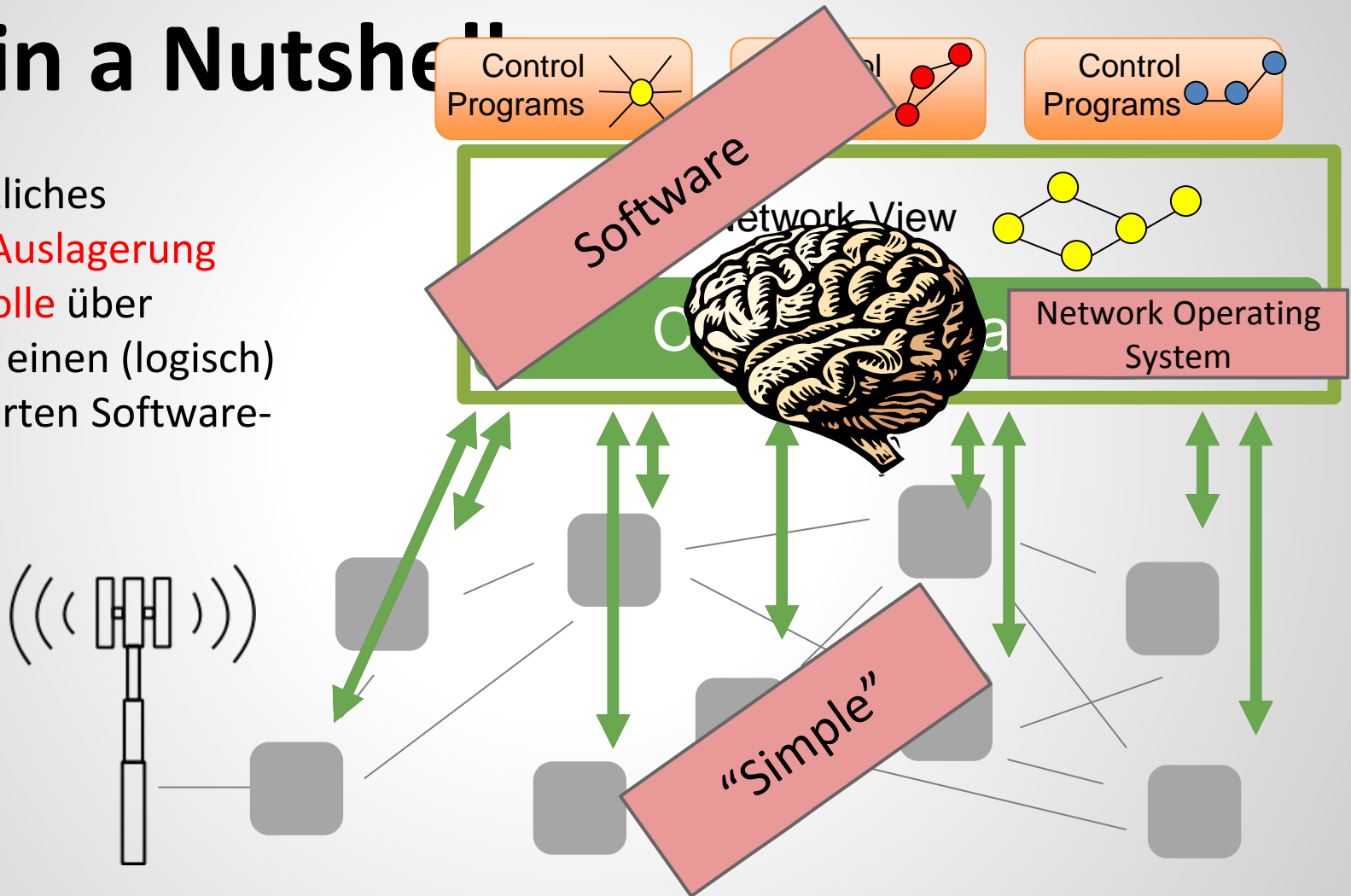


**Auslagerung!**



# SDN in a Nutshell

Grundsätzliches  
Konzept: **Auslagerung  
der Kontrolle** über  
Geräte an einen (logisch)  
zentralisierten Software-  
Kontroller

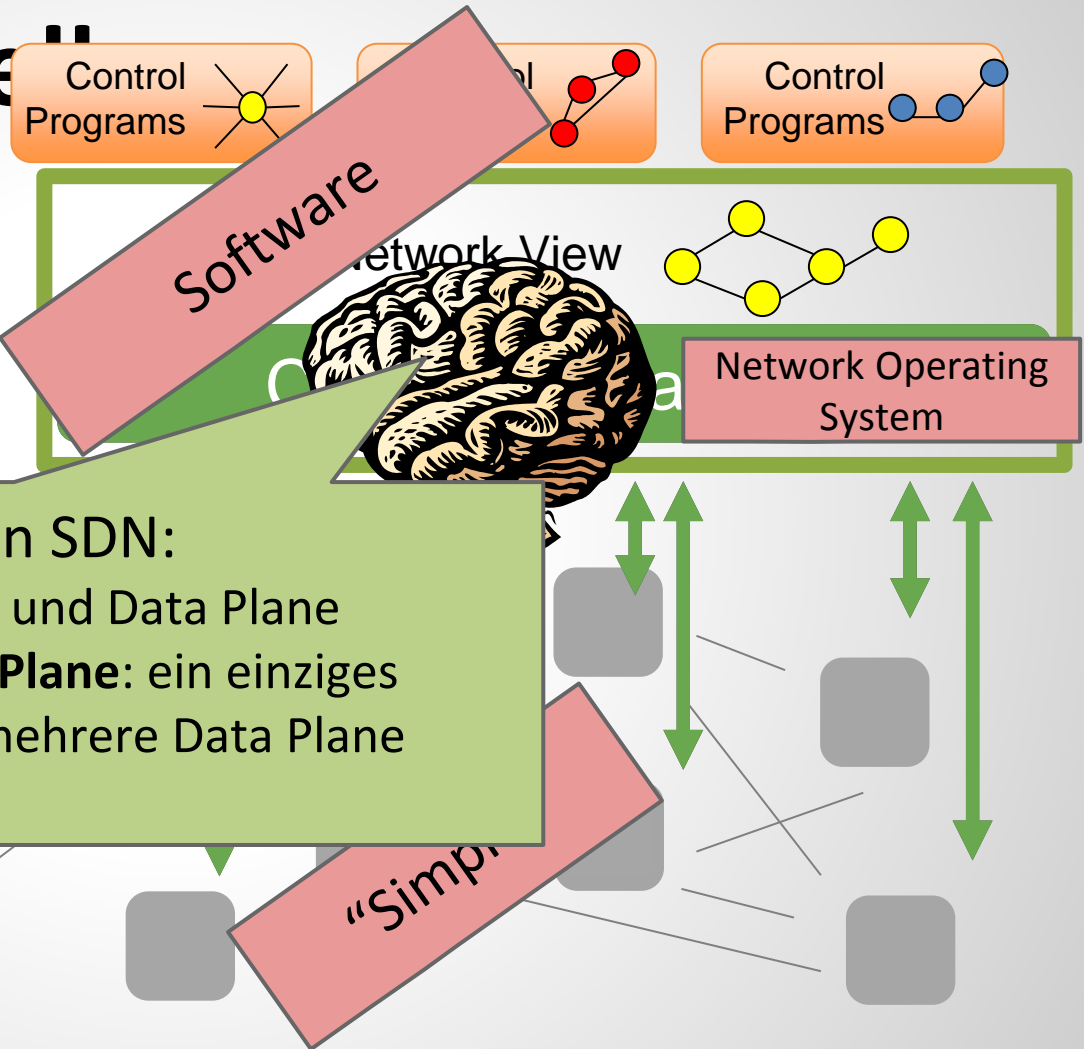


# SDN in a Nutshell

Grundsätzliches  
Konzept: **Auslagerung  
der Kontrolle** über  
Geräte an einen (logisch)  
zentralisierten Software

Die zwei Hauptkonzepte von SDN:

1. **Trennung** von Control Plane und Data Plane
2. **Konsolidierung der Control Plane**: ein einziges Kontrollprogramm steuert mehrere Data Plane Elemente



# SDN in a Nutshell

Vorteil: Entkoppelung! Control Plane kann sich unabhängig von der Data Plane

**weiterentwickeln:** Softwareinnovation oft schneller als Vendor-/Geräte-Innovation.

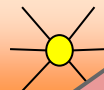
Und: kann auch versch. Kontrollsoftware mit versch. Vendor Geräten **kombinieren**.

Die zwei **Werkkonzepte** von SDN:

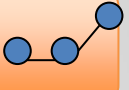
1. **Trennung** von Control Plane und Data Plane
2. **Konsolidierung der Control Plane:** ein einziges Kontrollprogramm steuert mehrere Data Plane Elemente

Vorteil: Einfacheres Netzwerkmanagement: eine inherent **nicht-lokale** Aufgabe. Und: vereinfacht **formale Verifikation**.

Control Programs

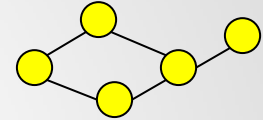


Control Programs

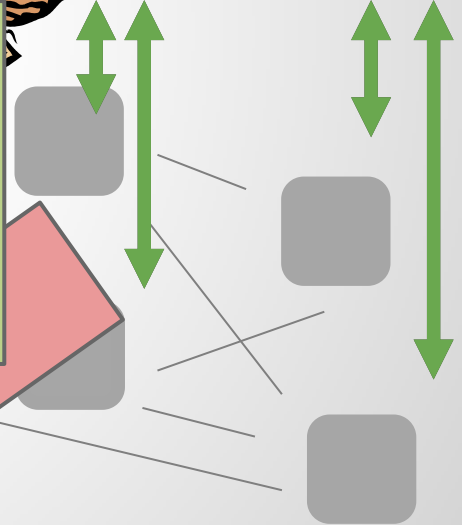


Software

Network View

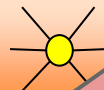


Network Operating System

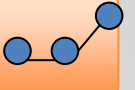


# SDN in a Nutshell

Control  
Programs



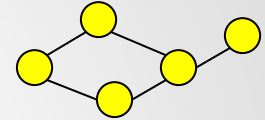
Control  
Programs



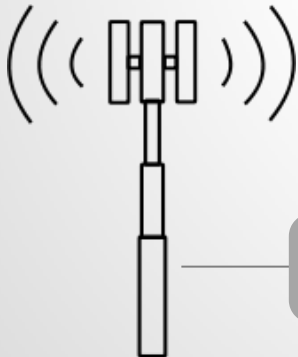
SDN bietet eine **klar-definierte Schnittstelle (API)** um Router, Switches, und andere Data Plane Elemente zu managen.

Wichtigste API heute: **OpenFlow**.

View



Network Operating  
System



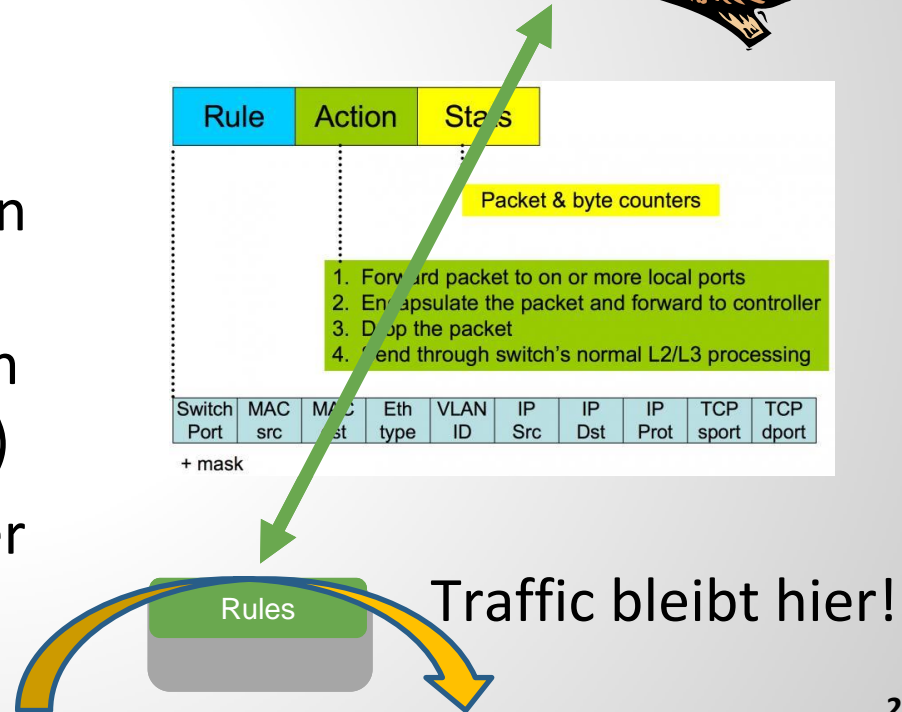
"Simple"

# Ein paar Worte zu OpenFlow



Wichtigste Prinzipien:

- **Flüsse nicht Pakete:** Basiert auf Regeln, die Flüsse definieren
- **Match-Action Paradigma:** Regeln matchen gewisse Pakete, und triggern entsprechende Aktionen (z.B. forward, drop, flood, count)
- **Flexibilität:** Matche L2-L4 Header

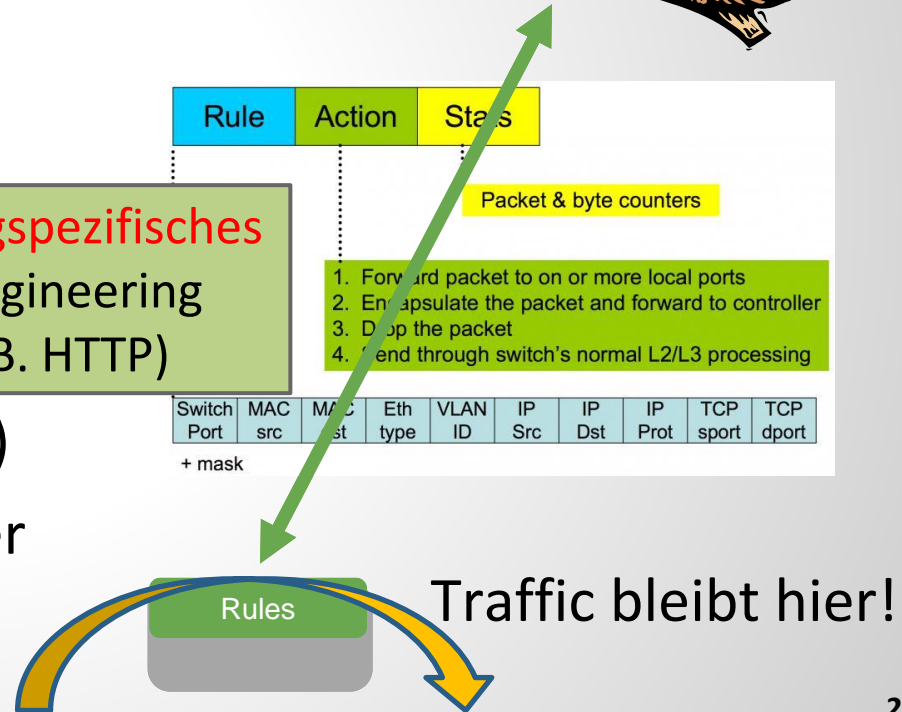


# Ein paar Worte zu OpenFlow



Wichtigste Prinzipien:

- **Flüsse nicht Pakete:** Basiert auf Regeln, die Flüsse definieren
- **Match-Action** Ermöglicht **anwendungsspezifisches Forwarding** / Traffic Engineering (abh. von TCP Ports, z.B. HTTP) (z.B. forward, drop, ... , count)
- **Flexibilität:** Matche L2-L4 Header



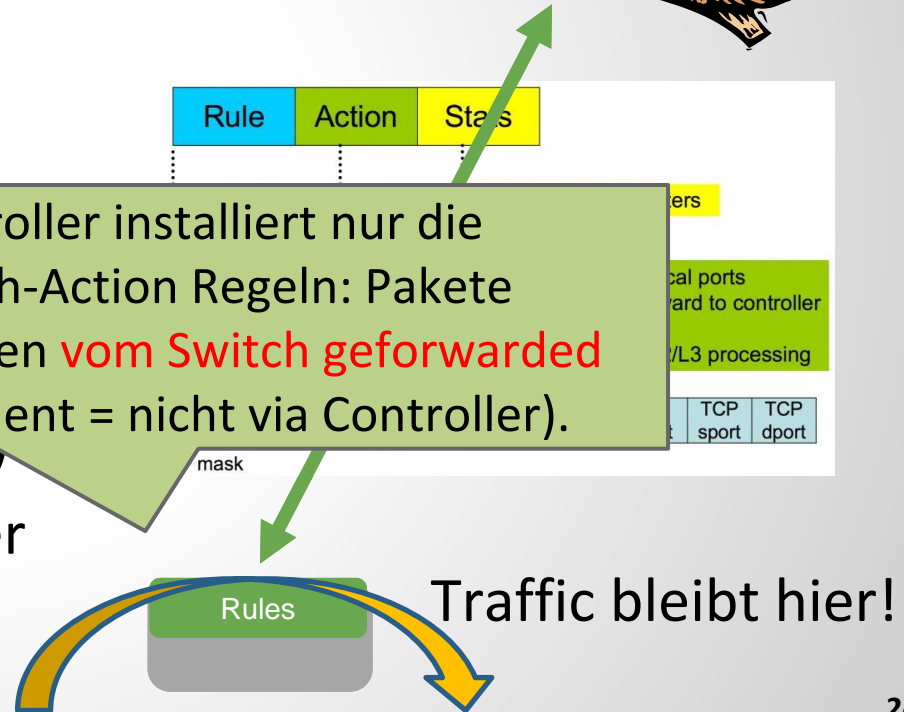
# Ein paar Worte zu OpenFlow



Wichtigste Prinzipien:

- **Flüsse nicht Pakete:** Basiert auf Regeln, die Flüsse definieren
- **Match-Action Paradigma:** Controller installiert nur die Regeln, die Matchen gewisse Pakete, und triggern entsprechende Aktionen (z.B. forward, drop, flood, count)
- **Flexibilität:** Matche L2-L4 Header

Kontroller installiert nur die Match-Action Regeln: Pakete werden **vom Switch geforwarded** (effizient = nicht via Controller).





# Ein paar Worte zu OpenFlow



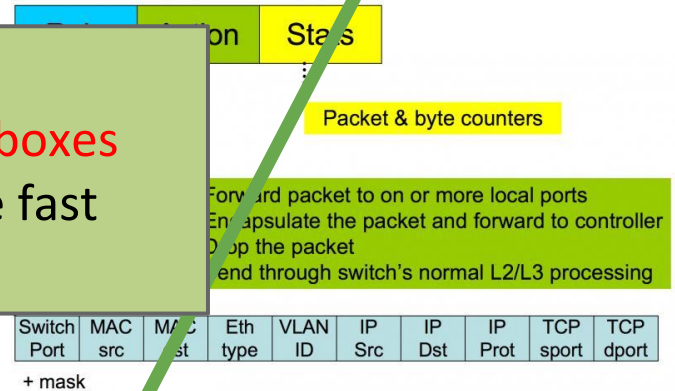
Wichtigste Prinzipien:

- **Flüsse nicht Pakete:** Basiert auf

Erlaubt es auch **einfache in-path Funktionen zu definieren** (können Router und einfache **Middleboxes wie NATs oder einfache Firewall** ersetzen: heute fast so viele Boxen wie Router!)

(z.B. forward, flood, count)

- **Flexibilität:** Matche L2-L4 Header



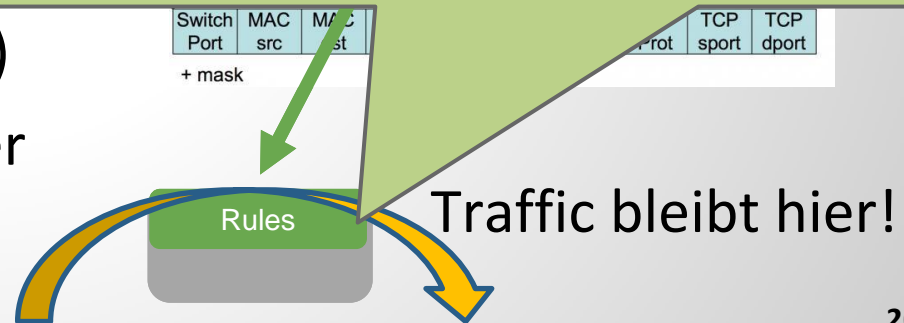
Traffic bleibt hier!

# Ein paar Worte zu OpenFlow



Wichtigste Prinzipien:

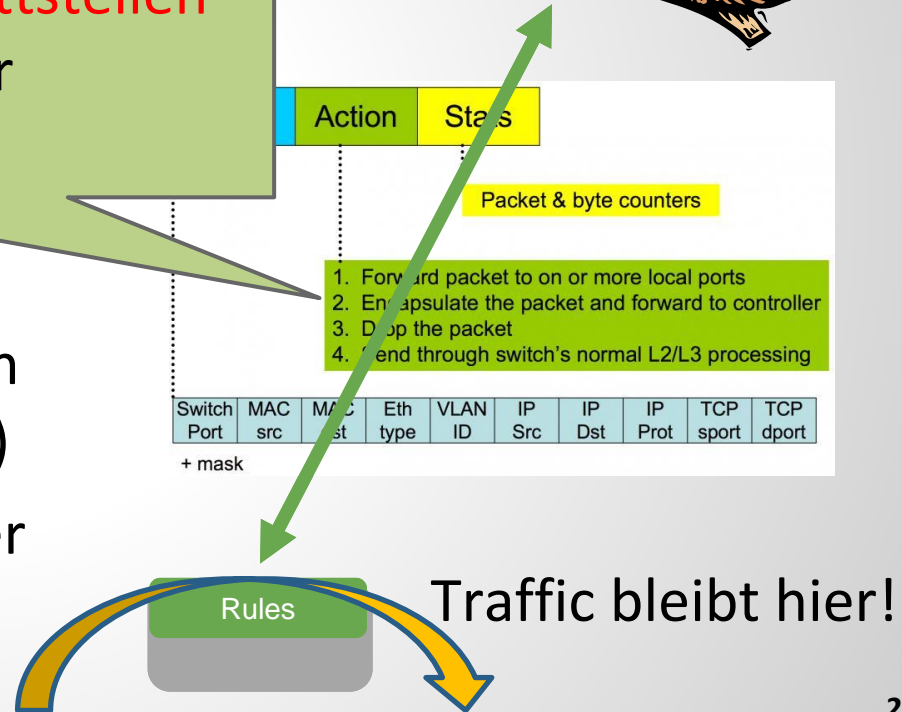
- **Flüsse nicht Pakete:** Basiert auf Regeln, die Flüsse steuern
- **Match-Action Paradigma:** Je nach installierten Regeln kann sich ein OpenFlow Switch verhalten wie ein **Router, Switch, Firewall, NAT**, oder etwas dazwischen! (Aber kein Packet Processing: dazu gibt's NFV.)  
matchen gewisse Header, die dann Aktionen triggern entsprechen (z.B. forward, drop, flood, count)
- **Flexibilität:** Matche L2-L4 Header



# Ein paar Worte zu OpenFlow

Es geht um **Offenheit**: Trennung von Control und Data Plane gibt's schon lange (z.B. Telefonnetz). Aber erst **offene Schnittstellen** ermöglichen Innovation in Controller Plattform und deren Anwendungen.

- **Match-Action Paradigma.** Regeln matchen gewisse Pakete, und triggern entsprechende Aktionen (z.B. forward, drop, flood, count)
- **Flexibilität:** Matche L2-L4 Header



# Fin naa Worte zu OpenFlow

Das Ziel: **Verallgemeinerung!**

- ... von **Geräten** (Switches, Router, Middleboxes)
- ... von **Routing** (nicht nur destination-based)
- ... von **Fluss Installation**: Grobe Regeln und **Wildcards**, **Proactive vs Reactive**, etc.
- Bietet Anwendung oder Benutzer eine flexible und logische **Netzwerksicht**

triggern entsprechende Aktionen  
(z.B. forward, drop, flood, count)

- **Flexibilität**: Matche L2-L4 Header

Viele Vendors unterstützen heute die OpenFlow API.



counters

more local ports  
and forward to controller

4. ... end through switch's normal L2/L3 processing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport
-------------	---------	---------	----------	---------	--------	--------	---------	-----------	-----------

+ mask

# Es ist eigentlich Zeit für SDN!

SDN = logische Konsequenz von allgemeinem Trend?

- Internet hat sich radikal geändert seit seinen Ursprüngen: Neue Bedürfnisse machen **Traffic Engineering (TE) immer komplexer** (z.B., Tunneling, MPLS, etc.)
- Aufkommen von Public Clouds (aka «killer app for SDN»): bedarf **Netzwerkvirtualisierung (NV)** und Isolation

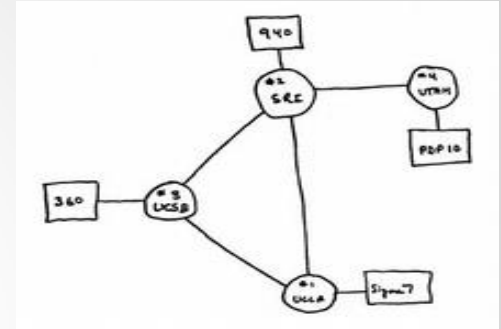
# Bedarf an flexiblerem Traffic Engineering

Das Internet:

**Urspr. Ziel:** Konnektivität zw. wenigen “Super-Computern”

**Urspr. Anwendungen:** File Transfer und Emails für Forscher

**Situation heute:** Nicht vernachlässigbarer Anteil der Weltbevölkerung ist rund um die Uhr online



## Neue Anforderungen:

- Mehr Verkehr, neue Anforderungen an Zuverlässigkeit und Predictability
- Deshalb: Infrastruktur muss effizienter genutzt werden, nutze auch Funktionen **innerhalb des Netzes** (z.B. Cache), **nicht nur destination-based** Routing, etc.
- Viele neue Anwendungen: Google Docs vs Datacenter Synchronisation vs On-demand Video: Anforderungen!
- Ausserdem: User mobility, IP Subnet Mobility, etc.



# Bedarf an flexiblerem Traffic Engineering

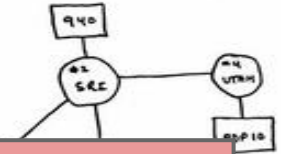
Das Internet:

**Urspr. Ziel:** Konnektivität zw. wenigen “Super-Computern”

**Urspr**

**Situat**

**Weltb**



## Vision der Netzwerkvirtualisierung:

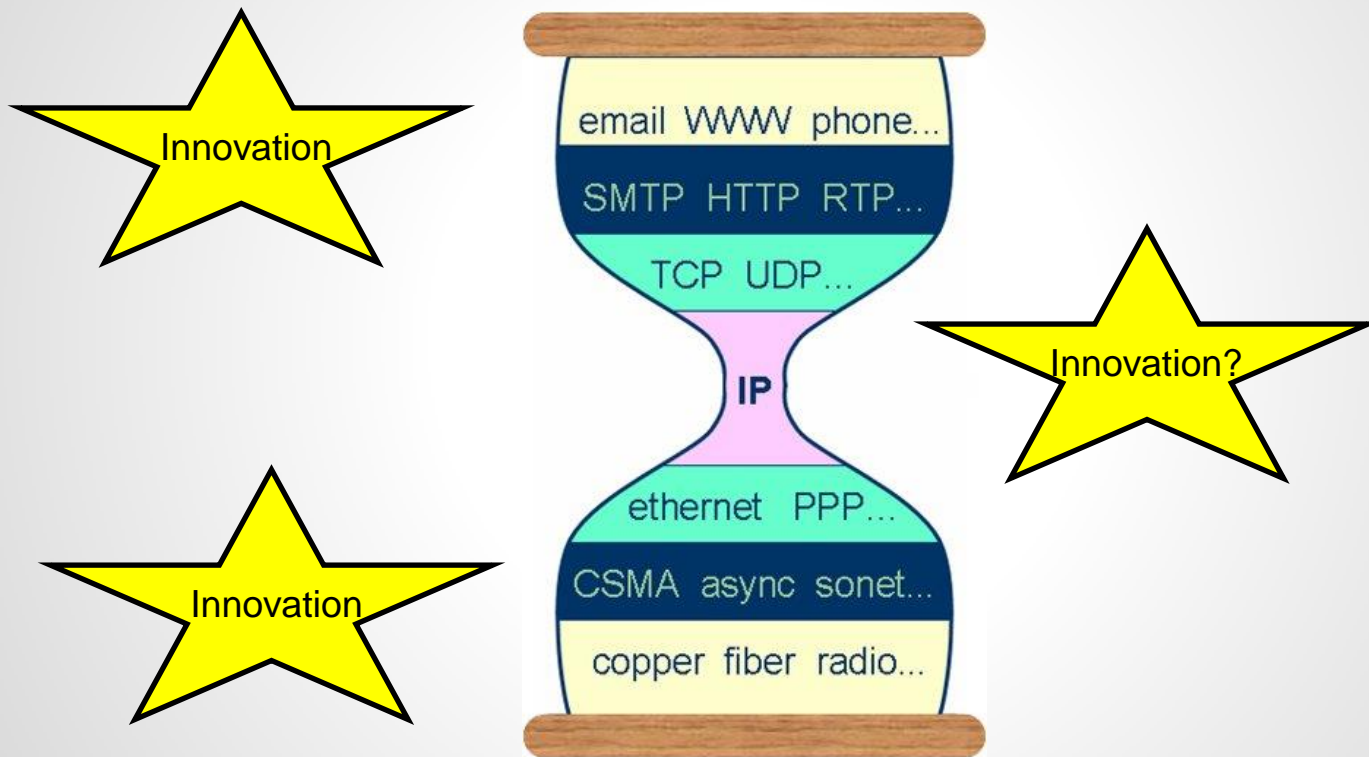
Auf Anwendung zugeschnittene virtuelle Netzwerke, teilen sich Infrastruktur, aber garantieren (Performance-) Isolation.



und Predictability

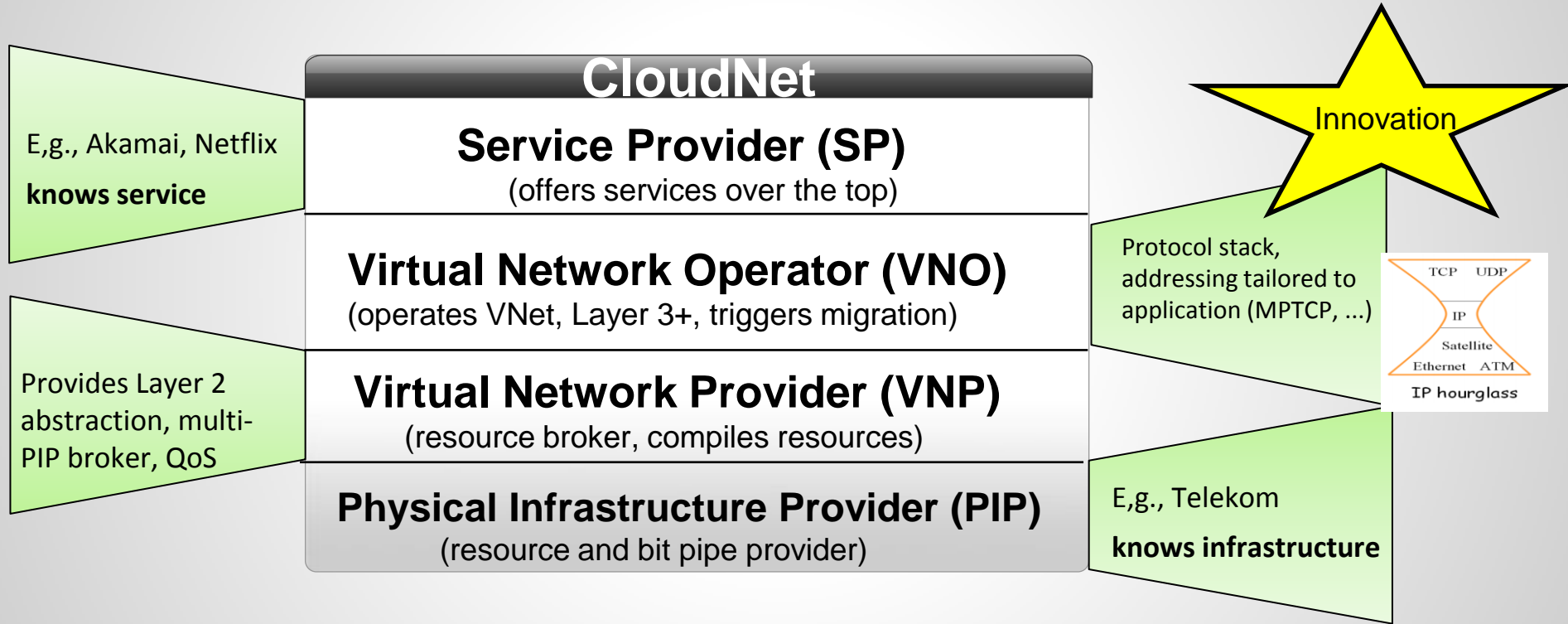
- Deshalb: Infrastruktur muss effizienter genutzt werden, nutze auch Funktionen **innerhalb des Netzes** (z.B. Cache), **nicht nur destination-based** Routing, etc.
- Viele neue Anwendungen: Google Docs vs Datacenter Synchronisation vs On-demand Video: Anforderungen
- Ausserdem: User mobility, IP Subnet Mobility, etc.

# “Ossification”



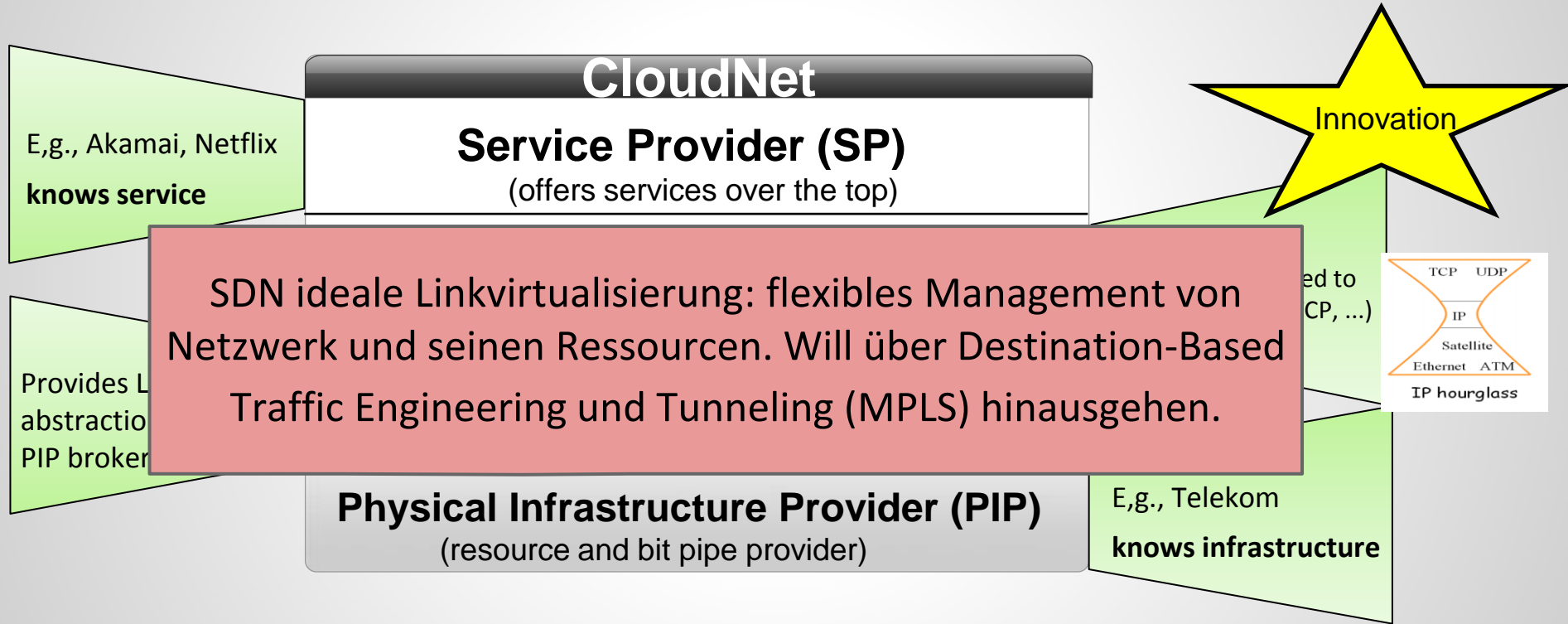


# Beispiel: CloudNets Prototyp am T-Labs



Heutiges Internet: In der Zukunft nur ein VNet von vielen?

# Beispiel: CloudNets Prototyp am T-Labs



Heutiges Internet: In der Zukunft nur ein VNet von vielen?

***Give me a break! Wirklich eine gute Idee?!***



# Ist das kein Rückschritt?!



- Paket-Switching und verteilte Kontrolle machten Netzwerke **robust** «gegen die Bombe»

- **Verteilte Control Planes** und **Paket-Switching** hatten einen Grund: “On Distributed Communications” (**Baran, 1964**)

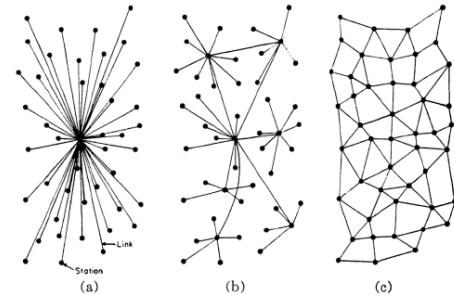


Fig. 1—(a) Centralized. (b) Decentralized. (c) Distributed networks.

1. Node & Edge Destruction
2. Distributed Routing

# Well, much happened since then...



# Zeit für Rückkehr zur zentralisierten Kontrolle?



**Zeit, darüber zu sprechen  
was SDN *nicht* ist! 😊**

# Was SDN *nicht* ist! (1)

- SDN ist nicht zwangsweise *physikalisch* zentralisiert, sondern nur *logisch*
  - Kontroller **können und sollen verteilt sein** (z.B. ONIX oder STN)
  - Gründe für verteilte Control Plane: für **Performanz**, Skalierbarkeit und **Robustheit**, aber auch versch. **administrative Zonen**
  - Z.B.: Häufige Data Plane Ereignisse sollten nah an den Data Plane Geräten behandelt werden!



# Was SDN *nicht* ist! (1)

- SDN ist nicht zwangsweise *physikalisch* zentralisiert, sondern nur *logisch*

- Kontroller **können und sollen verteilt sein** (z.B. ONIX oder STN)
- Gründe für verteilte Control Plane: **Performanz**, Skalierbarkeit und **Robustheit**, aber auch versch. **admin. Zonen**
- Z.B.: Häufige Data Plane Ereignisse müssen lokal behandelt werden!

In der Tat: Einige Funktionalität bleibt in der Data Plane (z.B., fast failover). Aber: eine verteilte und entkoppelte Kontrolle ist nicht trivial. Siehe später!

# Was SDN *nicht* ist! (2)

SDN ist nicht *active networking*

- Ähnliche Vision wie SDN: schnellere **Innovation**, fein-granulare und **einheitliche** Kontrolle über Geräte und Middleboxes
- Aber:
  - ANs sind ein **Clean Slate** Ansatz, und **Innovation in der Data Plane** schwieriger als in der Control Plane
  - ANs werden von **End-Usern und Forschern programmiert**, SDNs von **Admins**
  - ANs sind nicht backward-compatible (kein klarer **Migrationspfad**)
  - Anders als das OpenFlow Match-Action Paradigma, sind Active Networks schwierig zu **verifizieren**

# Was SDN *nicht* ist! (3)

- Aber vorallem: SDN ist nur ein *Werkzeug, keine Lösung!*
- SDN sagt weder wie die Control Plane designed werden soll noch löst SDN irgendein spezifisches Problem
- SDN bietet Forschern und Netzwerkadmins bloss eine *Plattform*

# Was SDN *nicht* ist! (3)

- Aber vorallem: SDN ist nur ein *Werkzeug, keine Lösung!*

- SDN soll noch **Aber zurück zu unserer Frage:** ed werden  
soll noch **Wieso SDN?** blem

- SDN bietet Forschern und Netzwerkadmins bloss eine *Plattform*

# Wo wird SDN heute eingesetzt?

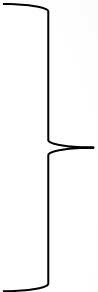
# Use Cases

Viele Anwendungen diskutiert heute, z.B. für:

- Unternehmensnetzwerke
- Rechenzentren
- Wide-Area Networks (WANs)
- IXPs
- ISPs

# Use Cases

Viele Anwendungen diskutiert heute, z.B. für:

- Unternehmensnetzwerke
  - Rechenzentren
  - Wide-Area Networks (WANs)
  - IXPs
  - ISPs
- 
- Bestehende Deployments!

Die Ursprünge von SDN: Z.B.  
Stanford Campus (*coined* "SDN")

Viele Anwendungen diskutiert heute, z.B. für:

- Unternehmensnetzwerke
- Rechenzentren
- Wide-Area Netzwerke (WANs)
- IXP
- ISP

Bestehende  
Deployments!

Killer Use Case heute?:  
Startups wie Nicira

Systeme von Google (B4) und  
Microsoft (SWAN)



Die Ursprünge von SDN: Z.B.  
Stanford Campus (*coined* "SDN")

Viele Anwendungen diskutiert heute, z.B. für:

- Unternehmensnetzwerke
- Rechenzentren
- Wide-Area Netzwerke (WANs)
- IXP
- ISP

Bestehende  
Deployments!

Killer Use Case heute?:  
Startups wie Nicira

Systeme von Google (B4) und  
Microsoft (SWAN)

Wie wird SDN genutzt?  
Wie wird SDN deployed?

Die Ursprünge von SDN: Z.B.  
Stanford Campus (*coined* "SDN")

3

Viele Anwendungen diskutiert heute, z.B. für:

- Unternehmensnetzwerke
- Rechenzentren
- Wide-Area Netzwerke (WANs)
- IXP
- ISP

Bestehende  
Deployments!

Killer Use Case heute?:  
Startups wie Nicira

1

Systeme von Google (B4) und  
Microsoft (SWAN)

2

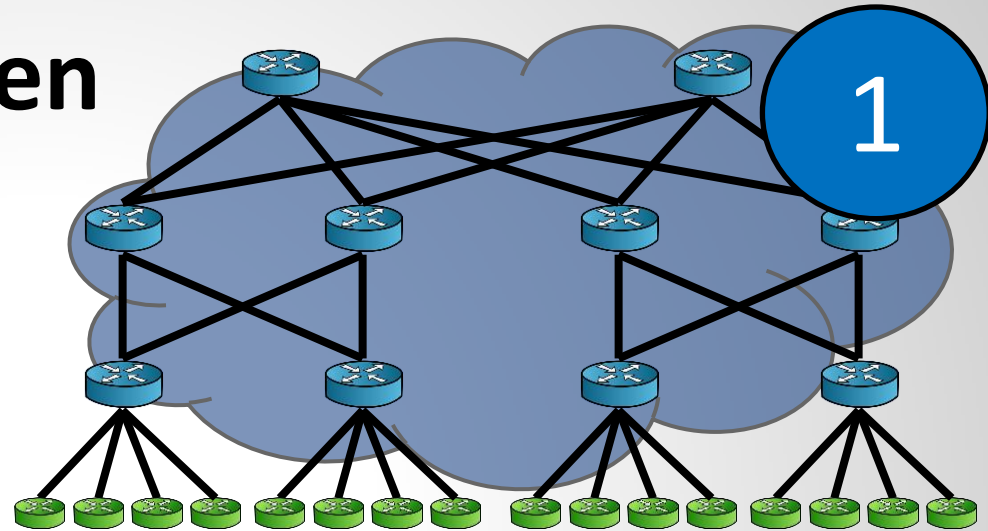
Wie wird SDN genutzt?  
Wie wird SDN deployed?

# SDN in Rechenzentren

## Charakteristiken

- Bereits stark virtualisiert
- Ziemlich homogen

## Wieso SDN?



Wie?

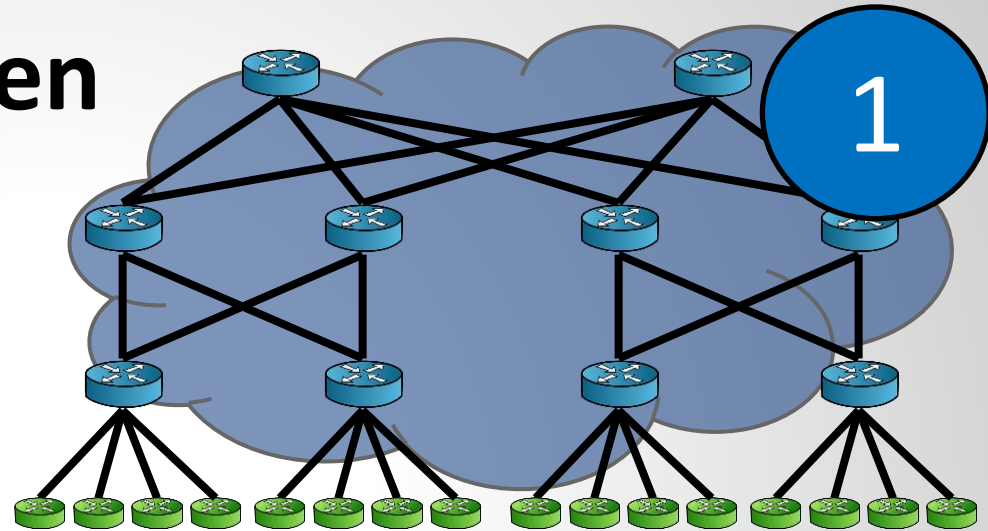
# SDN in Rechenzentren

## Charakteristiken

- Bereits stark virtualisiert
- Ziemlich homogen

## Wieso SDN?

- **Entkopplt** Anwendungen von phys. Infrastruktur
- Ermöglicht **Virtual Networks** (z.B., Nicira): eigene Adressräume für Tenants, Isolation, unterbruchfreie VM Migration, ...
- Performanz: **Durchsatz**



Wie?

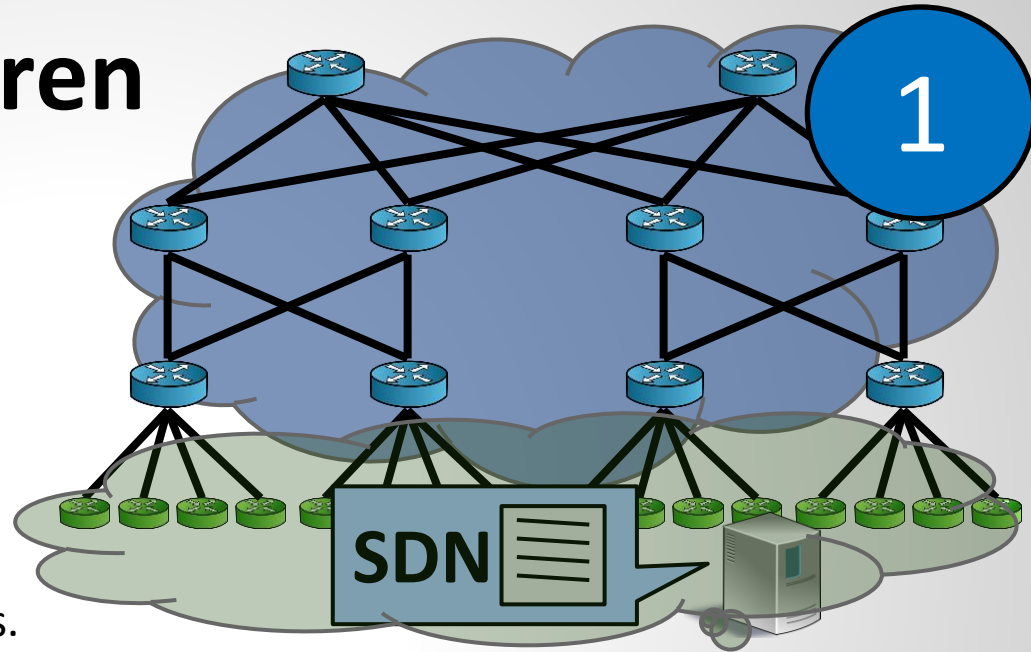
# SDN in Rechenzentren

## Charakteristiken

- Bereits stark virtualisiert
- Ziemlich homogen

## Wieso SDN?

- **Entkopplt** Anwendungen von phys. Infrastruktur
- Ermöglicht **Virtual Networks** (z.B., Nicira): eigene Adressräume für Tenants, Isolation, unterbruchsfreie VM Migration, ...
- Performanz: **Durchsatz**



Wie?

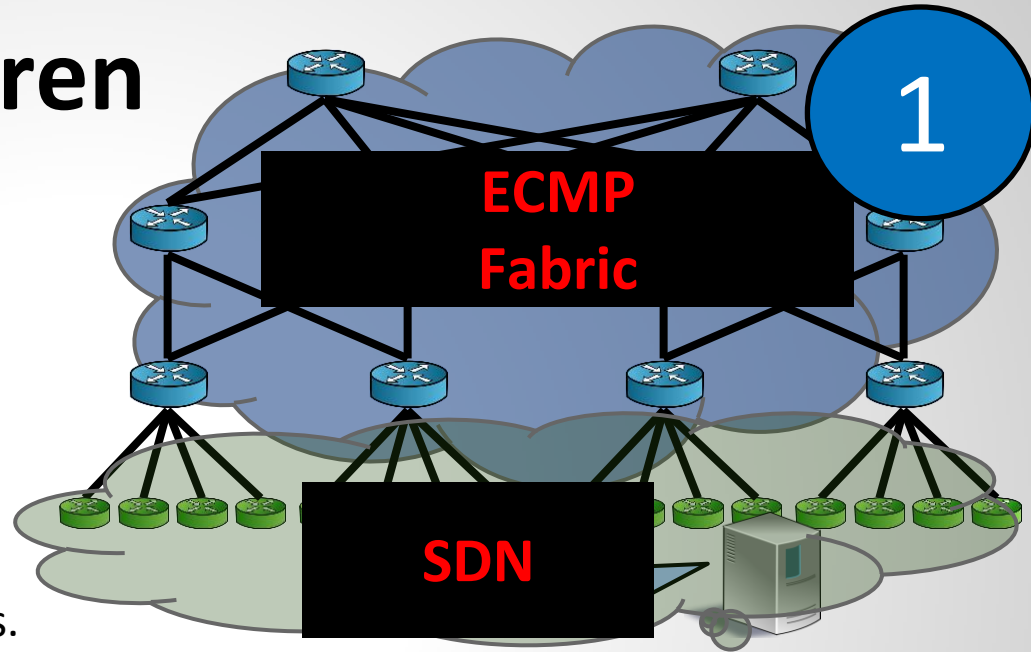
# SDN in Rechenzentren

## Charakteristiken

- Bereits stark virtualisiert
- Ziemlich homogen

## Wieso SDN?

- **Entkopplt** Anwendungen von phys. Infrastruktur
- Ermöglicht **Virtual Networks** (z.B., Nicira): eigene Adressräume für Tenants, Isolation, unterbruchfreie VM Migration, ...
- Performanz: **Durchsatz**



## Wie?

- Control Plane: geteilt in zwei unabh. Control Planes
  - Edge = Paketklassifizierung und Tunnel Endpunkte
  - Core = **Fabric Abstraction (ECMP)**
- **SDN Deployment einfach**: einfaches **Softwareupdate**, terminiere Links an Software Switches (Open vSwitch)

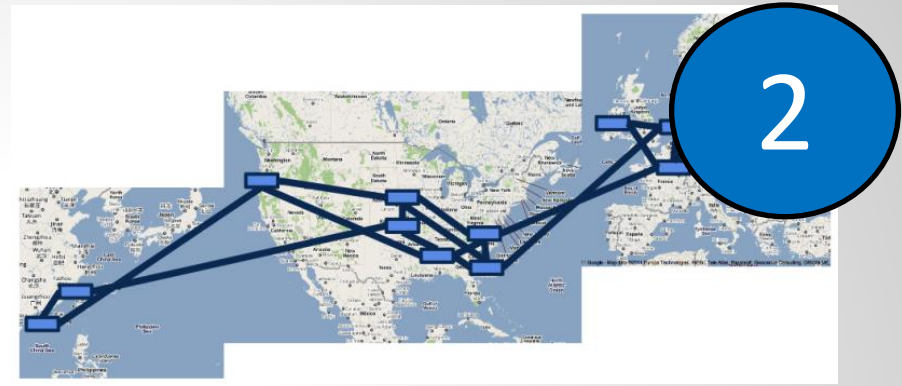
# SDN in WANs

## Charakteristik

- **Bandbreite** kostbar (WAN Traffic wächst schnell)
- **Kleine Deployments**: nicht viele Sites
- Viele untersch. **Anwendungen** und Anforderungen, Latenz wichtig

Wieso SDN?

Wie?



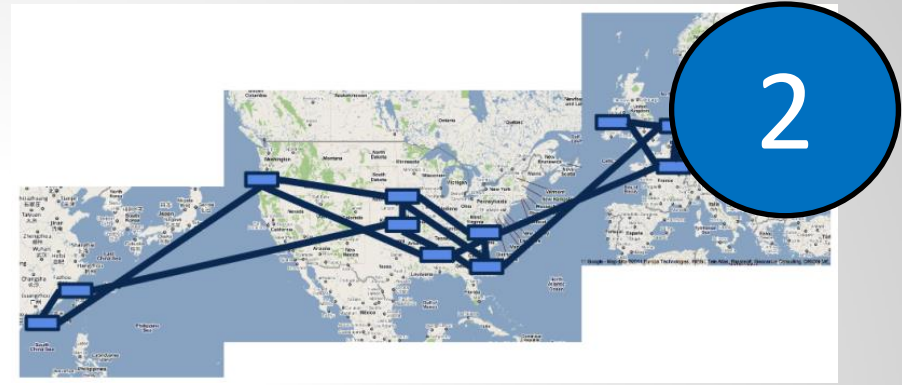
# SDN in WANs

## Charakteristik

- **Bandbreite** kostbar (WAN Traffic wächst schnell)
- **Kleine Deployments**: nicht viele Sites
- Viele untersch. **Anwendungen** und Anforderungen, Latenz wichtig

## Wieso SDN?

- Bessere **Utilization** (z.B., Google B4) und geringere **Kosten** (z.B., Microsoft SWAN)
- **Differenziere** Anwendungen (Google Docs Latenz-sensitiv vs Datacenter Synchronization)



Wie?



# SDN in WANs

## Charakteristik

- **Bandbreite** kostbar (WAN Traffic wächst schnell)
- **Kleine Deployments**: nicht viele Sites
- Viele untersch. **Anwendungen** und Anforderungen, Latenz wichtig

## Wieso SDN?

- Bessere **Utilization** (z.B., Google B4) und geringere **Kosten** (z.B., Microsoft SWAN)
- **Differenzieren** Anwendungen (Google Docs Latenz-sensitiv vs Datacenter Synchronization)



## Wie?

- Ersetze IP “core” Router (running BGP) **am Rand** des Rechenzentrums (Ende des **long-haul Fiber**)
- Inkrementelles Ersetzen der Router möglich (**inkrementelle Data Plane**)

# SDN in WANs

2

Charak

- Ba
- sch
- Kle
- Vie
- An

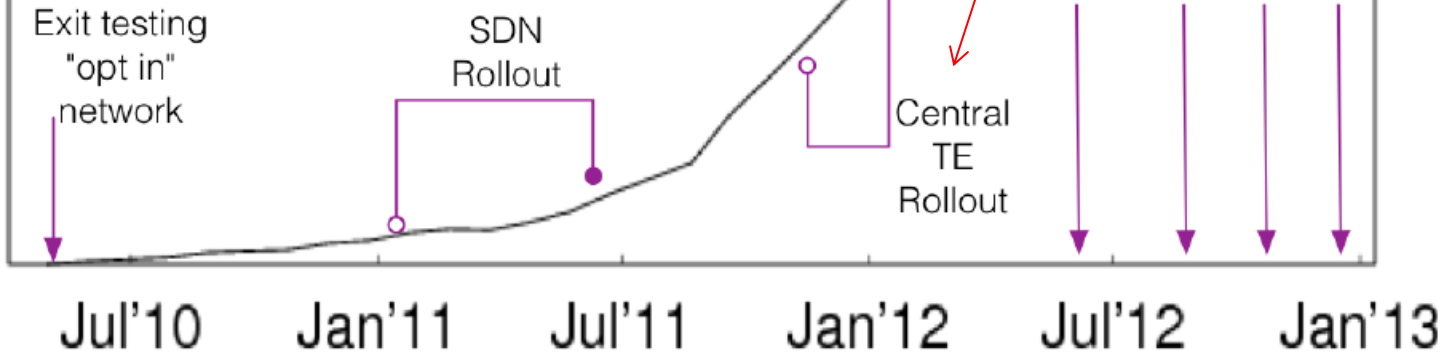
Traffic

Wieso

- Be
- B4
- Mi
- Dif
- (Go
- Datacenter Synchronization)

- a: Reduce Tunnel Ops by caching recently used tunnels
- b: Adapt TG modifies to unresponsive OFCs to reduce drops
- c: Link Coloring Based Path Selection
- d: Route flows differently based on QoS

Erste SDN Benefits erst  
am "Flag Day" (nur  
Data Plane  
inkrementell)

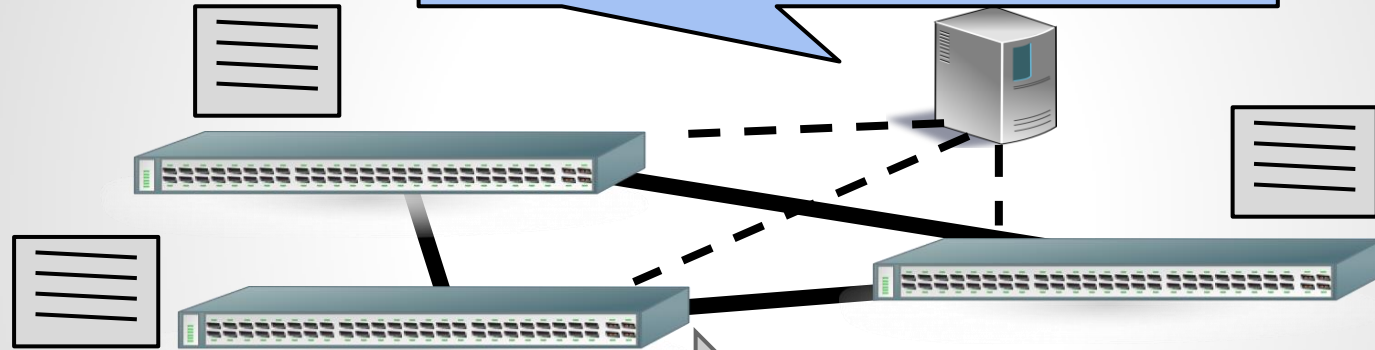


g

# Use Case: Wieso SDN im Unternehmen?

3

**Ziel:** einfaches Netzmanagement:  
zentral und programmatisch definierte  
Policies



Heute:  
verteilte Dateien

**Ziel**

SDN: zentral

**Vorteil: Zentrale Sicht, Automatisierung und Abstraktion von Netzwerken**

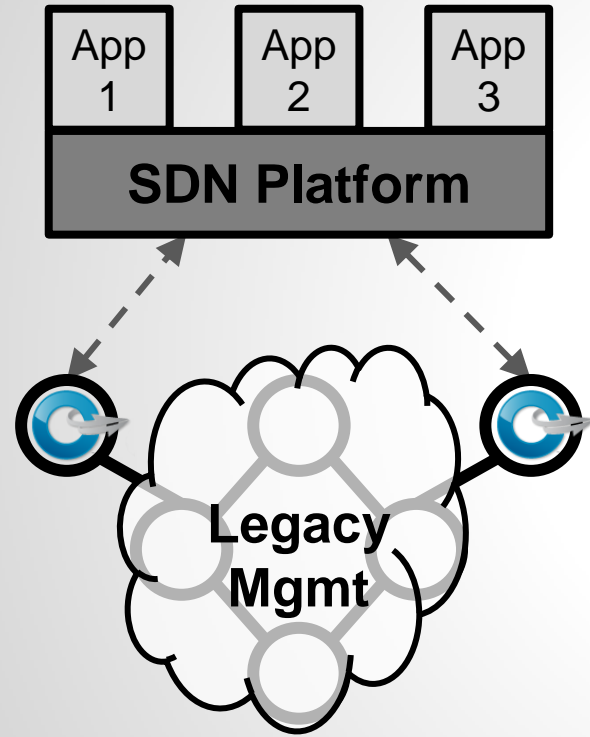
# SDN Deployment in Enterprise: Nicht-Trivial!

- Netzwerk häufig **organisch** gewachsen und komplex
- Zu managende Netze können **gross** sein...
- ... aber Infrastruktur-**Budgets** begrenzt

# SDN Deployment in Enterprise: Nicht-Trivial!

- Netzwerk häufig **organisch gewachsen und komplex**
- Zu managende Netze können **gross** sein...
- ... aber Infrastruktur-**Budgets begrenzt**
- Deshalb Frage:
  - Kann man SDN **inkrementell installieren**?
  - Z.B. auch am Edge? Wie im Datacenter?
  - Welche **SDN Benefits** können in hybridem Netzwerk genutzt werden?

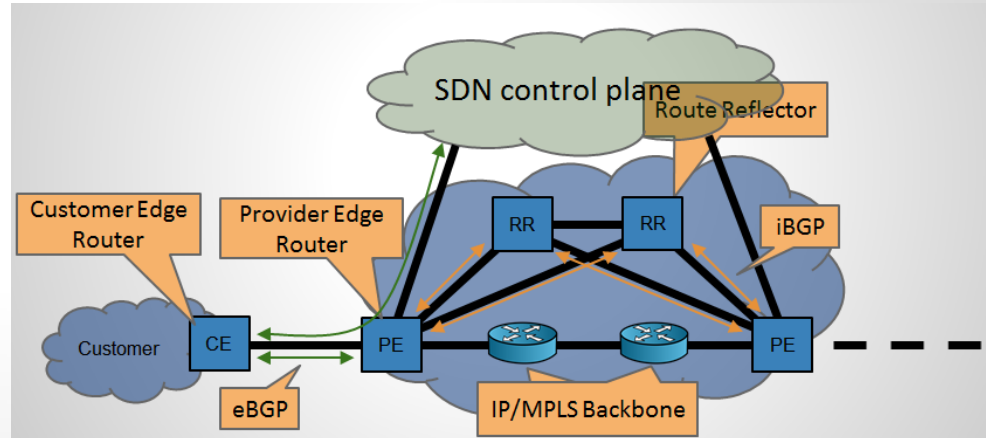
# Hybride Netzwerke Heute: Edge



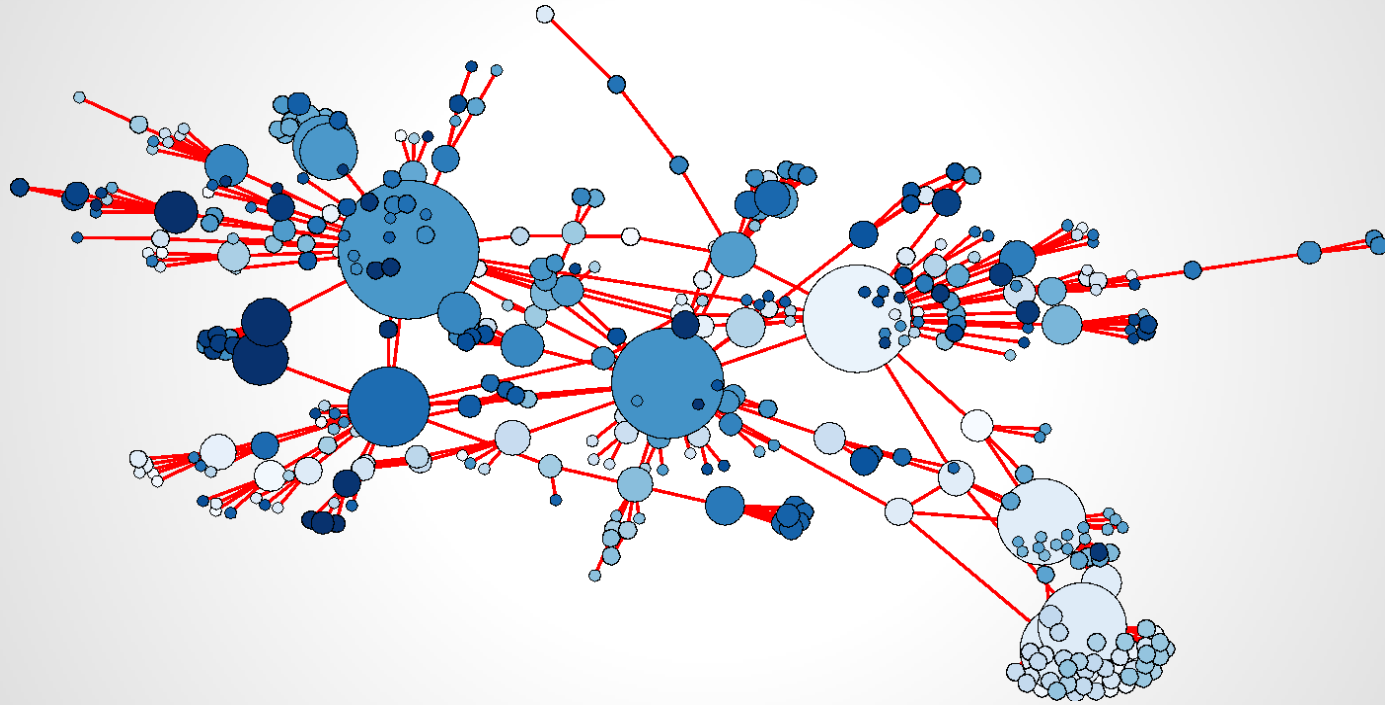
Edge-only Ansatz

Heute häufig am Edge deployed:

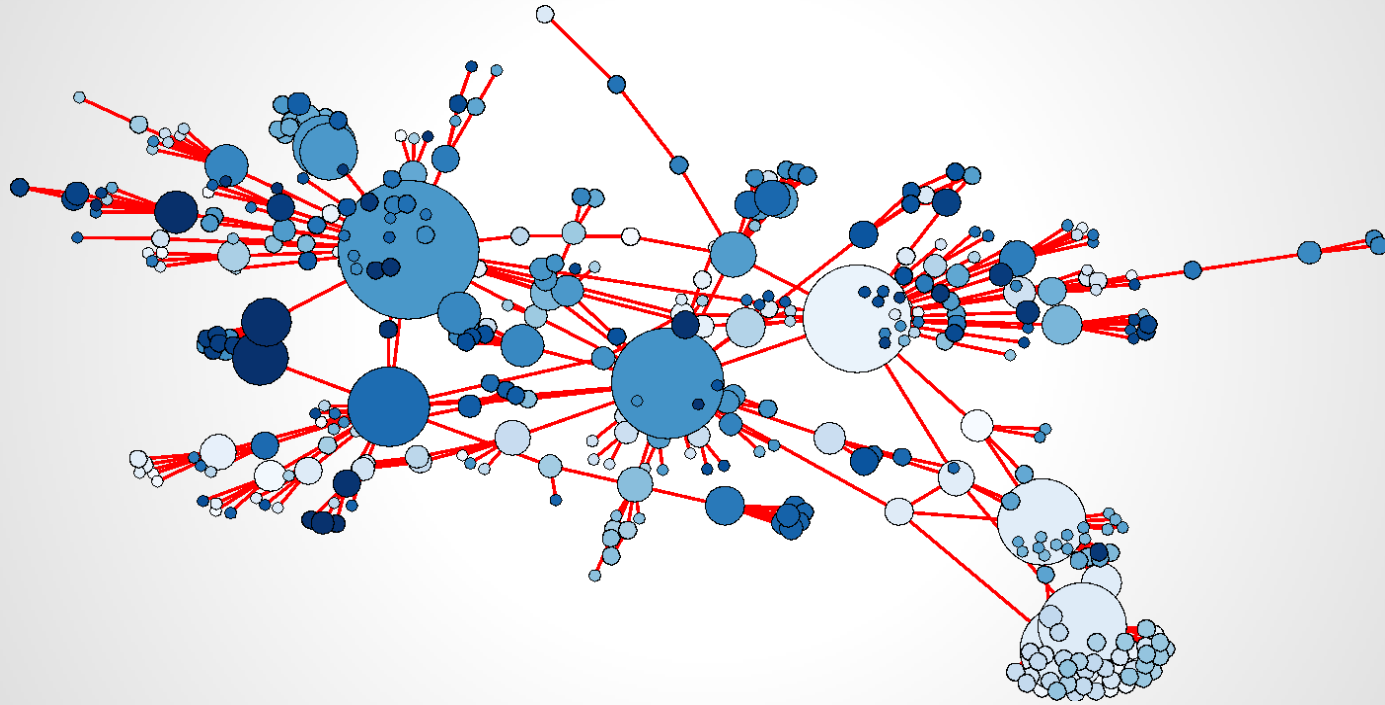
- Z.B. in **Rechenzentren**: Rand ist Software, und im Core reicht eine ECMP Fabric
- Z.B. **NTT Provider Edge**: BGP Funktionalität vom Edge Router zum Controller migriert



# Problem: Wo ist der Edge des Enterprise Netzwerks?!



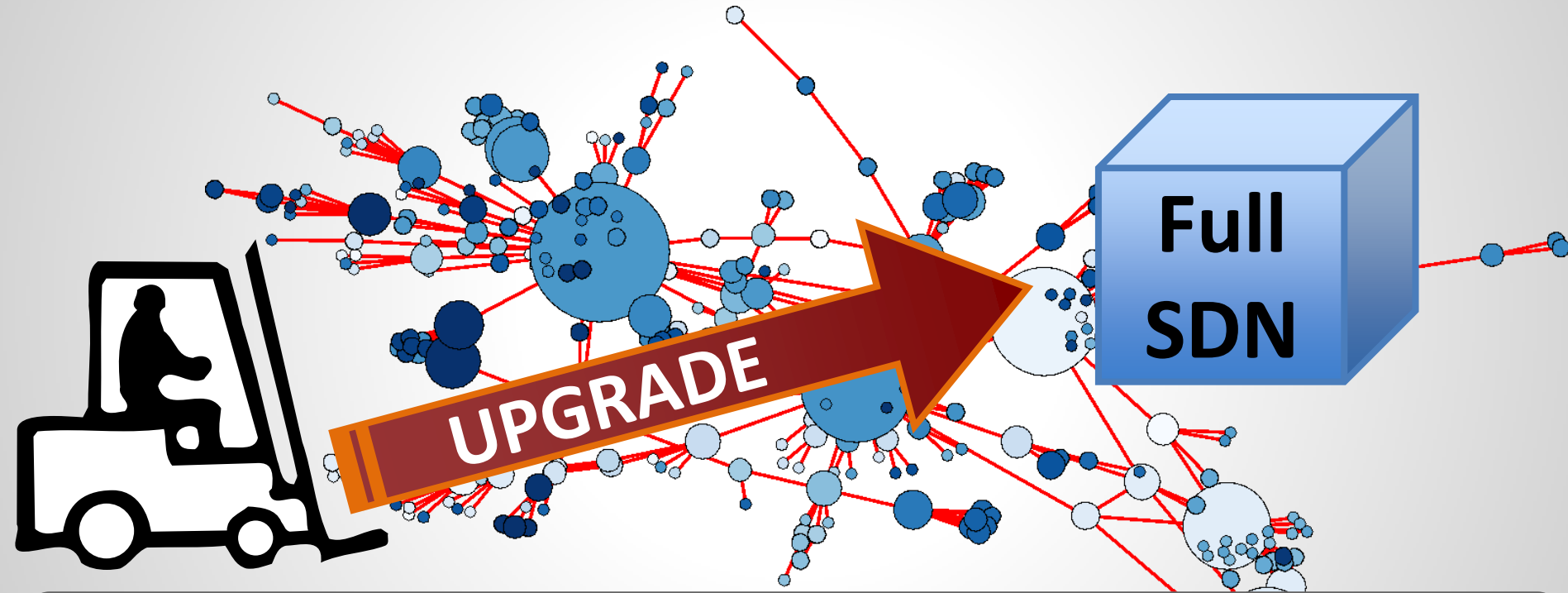
# Problem: Wo ist der Edge des Enterprise Netzwerks?!



Edge kann gross sein, und ist nicht in Software!



# Wir haben ein Deployment Problem...



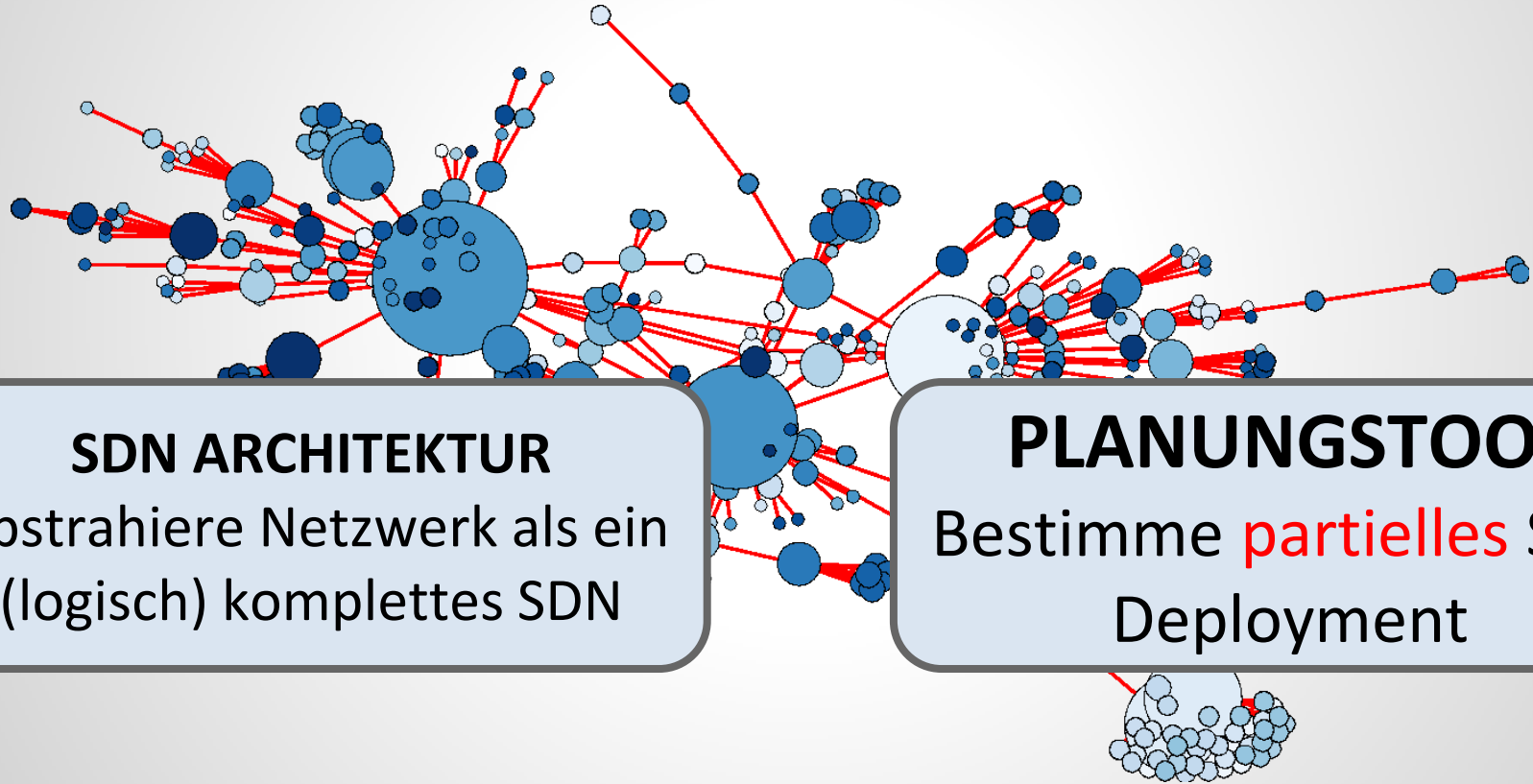
Teuer und unerwünscht: Will inkrementell upgraden, auch für **Confidence Building**! Will keinen “**Flag Day**” der Control Plane.

# Einsichten

1. Oft ist es gar nicht nötig, das Netzwerk vollständig zu upgraden!
2. Deployment muss nicht unbedingt am Rand sein: Deployment an “zentraler” Stelle oft besser

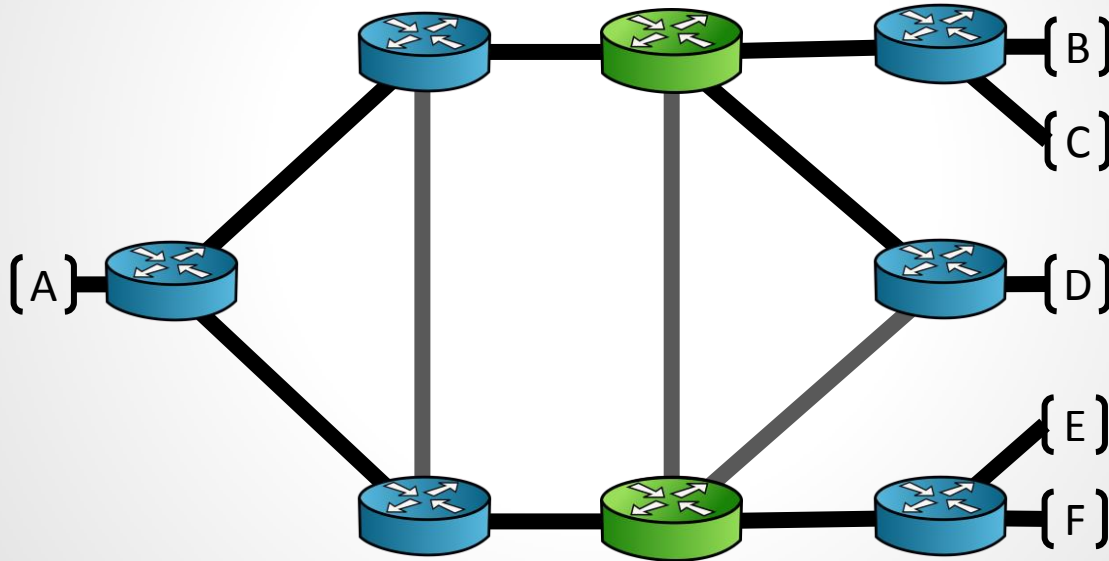
**Lösung: Panopticon!**

# Panopticon



# Partielles SDN Deployment ( )

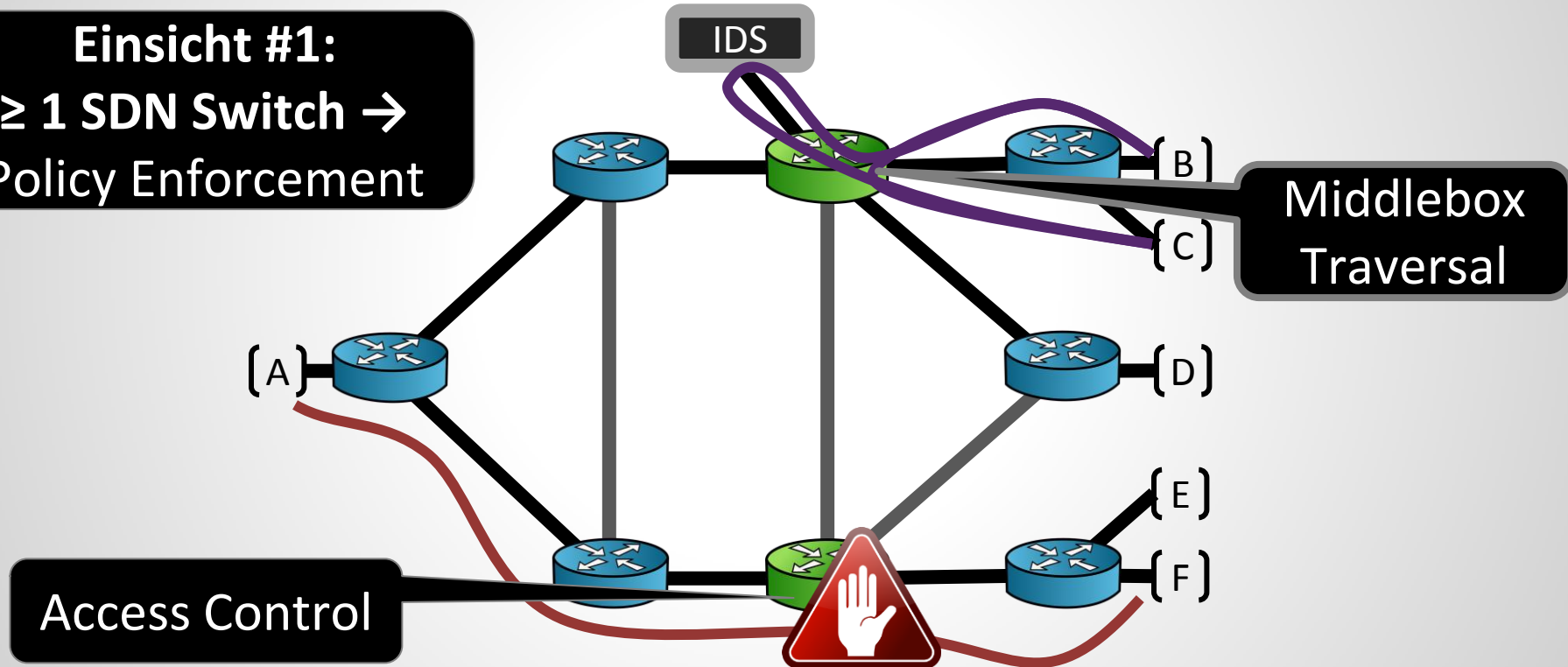
Gegeben oder geplant:



# *Eigentlich reicht ein einziger Switch...*

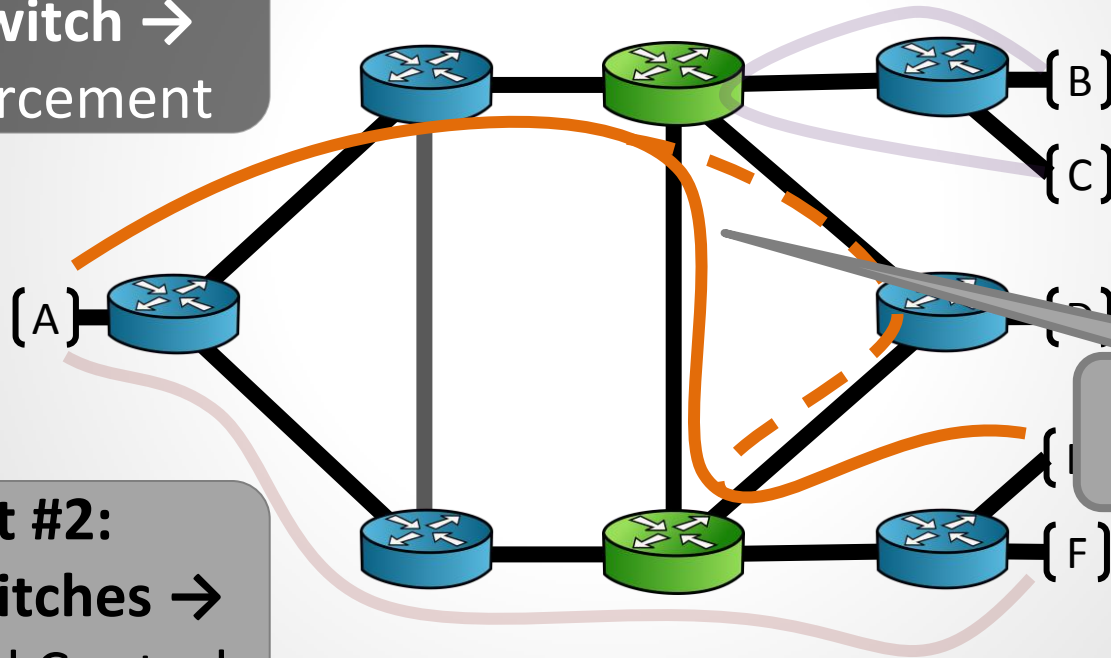
Kriege Funktionalität durch Waypoint Enforcement!

**Einsicht #1:**  
 $\geq 1$  SDN Switch  $\rightarrow$   
Policy Enforcement



# Mehr Deployment = Mehr Flexibilität

**Einsicht #1:**  
 $\geq 1$  SDN Switch  $\rightarrow$   
Policy Enforcement



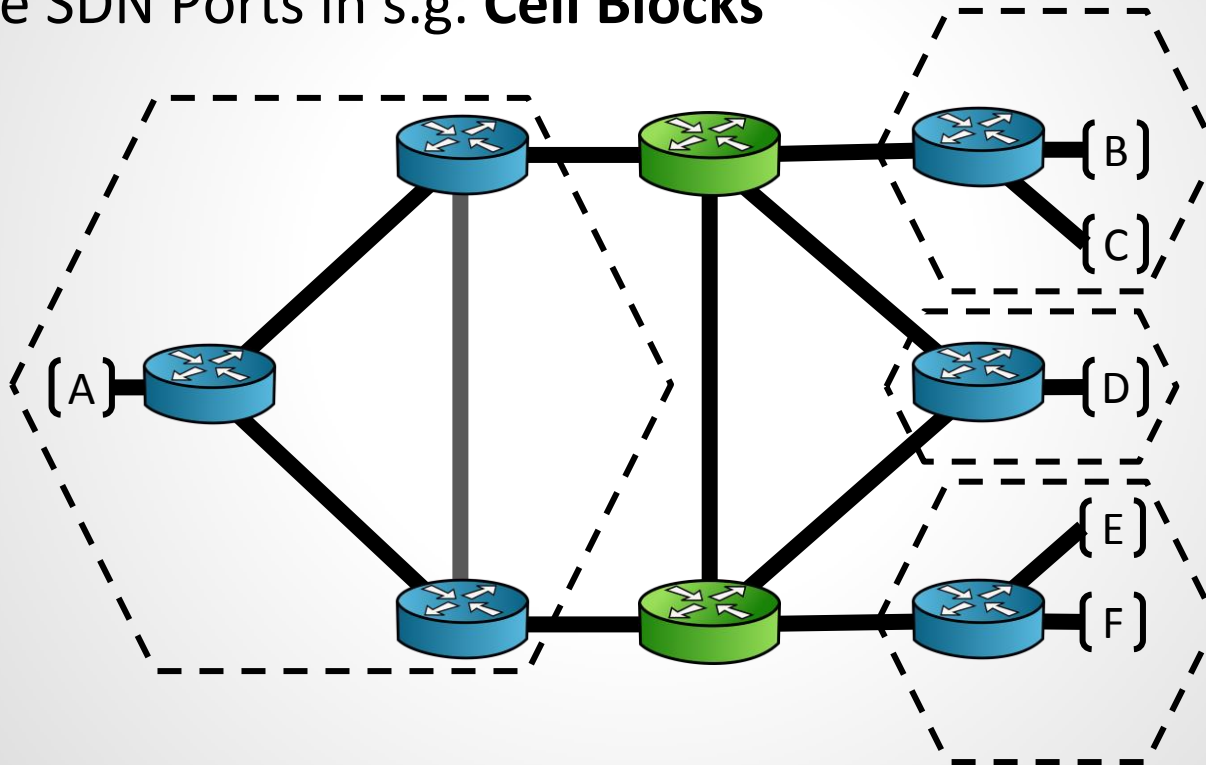
Traffic  
Load-balancing

**Einsicht #2:**  
 $\geq 2$  SDN Switches  $\rightarrow$   
Fine-grained Control

# Panopticon: Bietet Applikation Logical-SDN Abstraktion

Konstruktion:

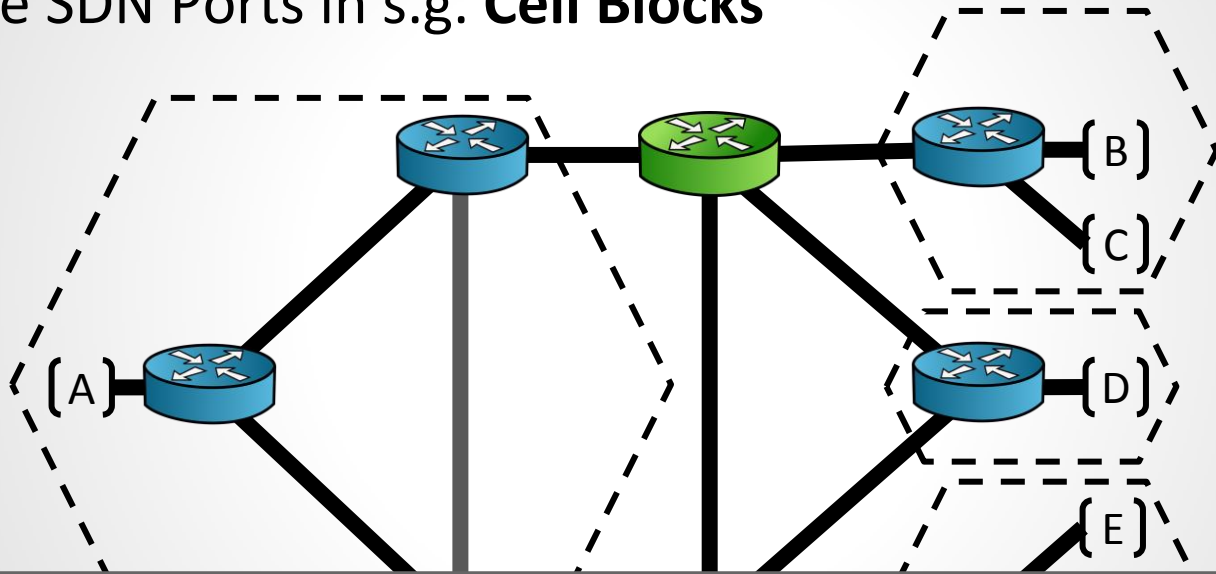
1. Gruppieren SDN Ports in s.g. **Cell Blocks**



# Panopticon: Bietet Applikation Logical-SDN Abstraktion

Konstruktion:

1. Gruppieren SDN Ports in s.g. **Cell Blocks**

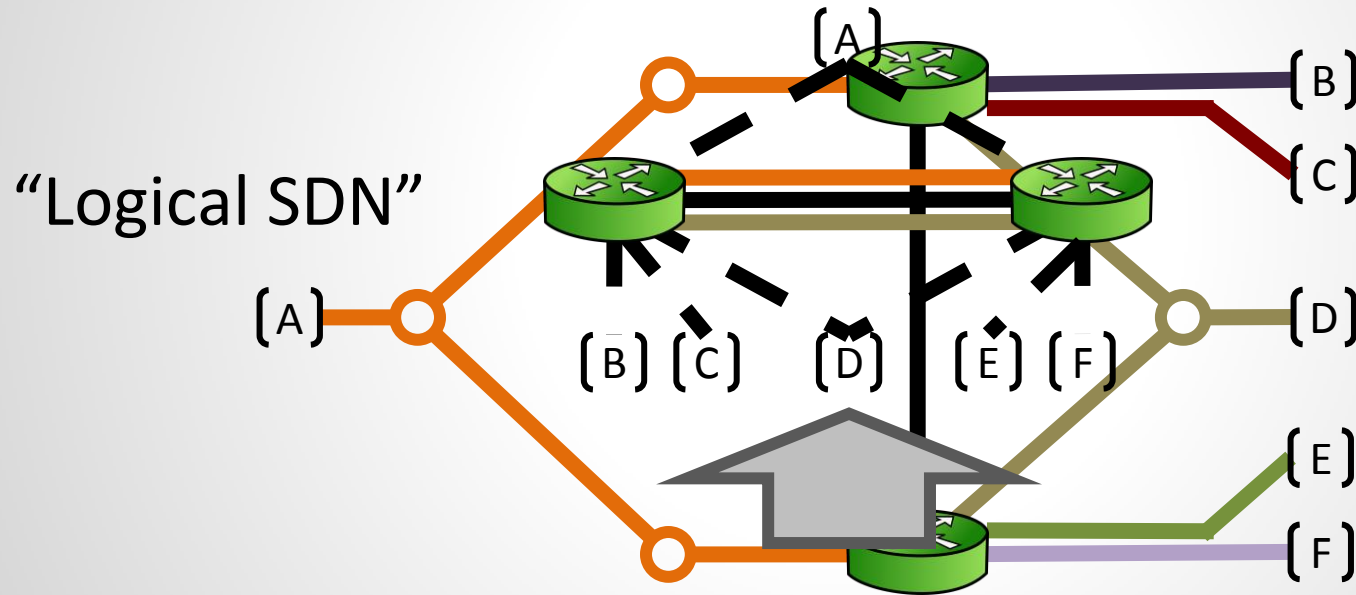


**Per-port Spanning Trees garantieren  
Waypoint Enforcement**



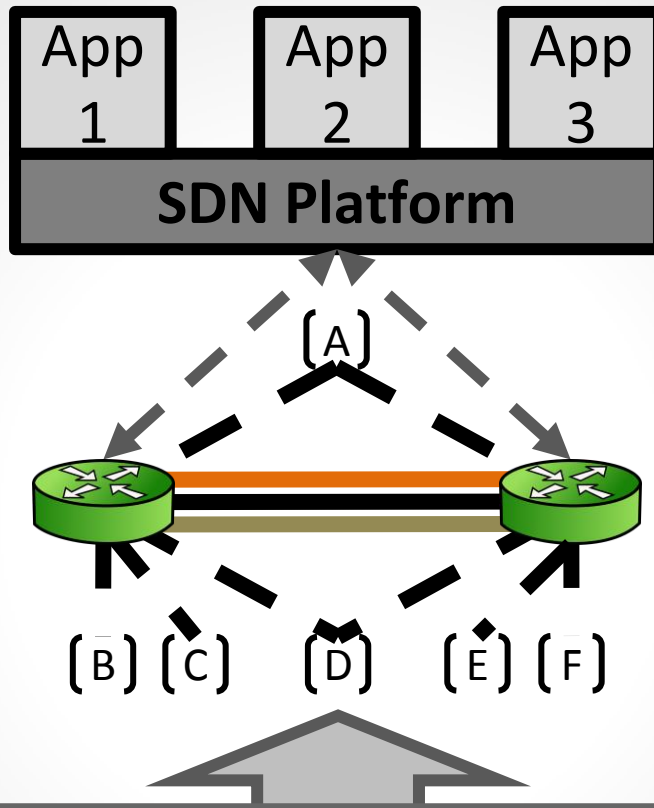
# Panopticon: Bietet Applikation Logical-SDN Abstraktion

## 2. Bilde per SDN-Port VLANs



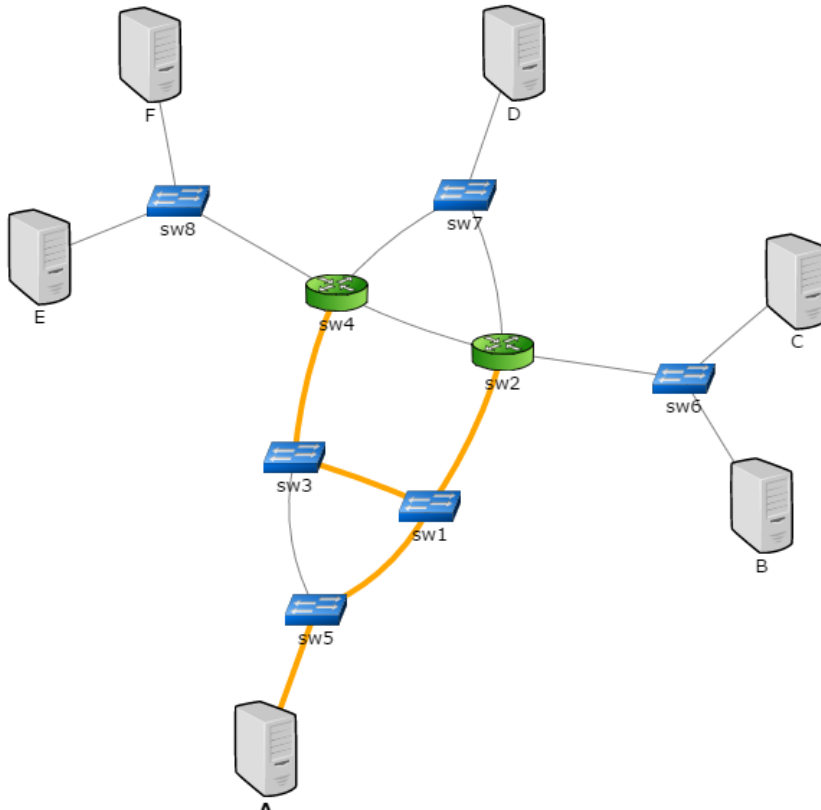
### 3. Bilde logische Abstraktion

“Logical SDN”



**PANOPTICON bietet Abstraktion von (fast)  
vollständigem SDN in einem partiell upgegradeten Netzwerk!**

# Beispiel



Physical

Logical

- Note whether any changes have appeared in the flow table of **sw2**.
- Look again at the MAC table of switch **sw6**; observe the VLANs used by the traffic traversing this switch

## Challenge

What is the designated SDN switch for the source-destination pair **B** and **C**?

## Hints

To view a flow table, select an SDN switch in the network topology and click on **Table** in the bottom action bar.

Look at the logical network view.

## Take-Aways

All traffic between **B** and **C** must traverse an SDN switch because **B** and **C** are connected to the network via SDN controlled ports.

You are now ready to move to the next task.

sw2

Every 1.0s: ovs-ofctl dump-fl... Fri Sep 19 18:25:37 2014

NXST\_FLOW reply (xid=0x4):

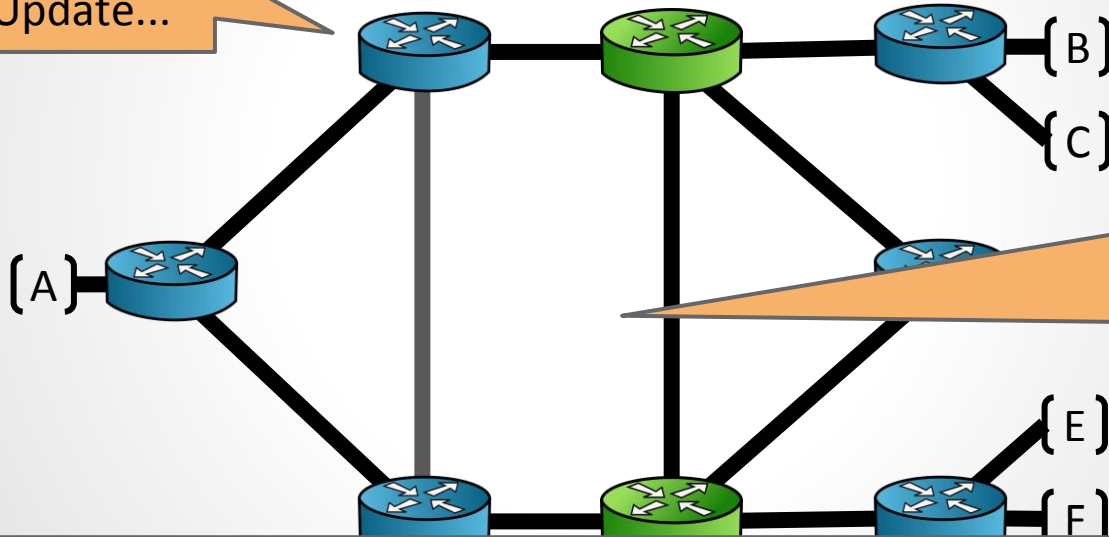
```
cookie=0x0, duration=269.257s, table=0, n_packets=263, n_bytes=25546, idle_age=11, in_port=2, dl_src=00:00:00:00:00:02, dl_dst=00:00:00:00:00:03 actions=mod_vlan_vid:4, IN_PORT
cookie=0x0, duration=388.481s, table=0, n_packets=9, n_bytes=890, idle_age=380, in_port=4, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:04 actions=mod_vlan_vid:7, output:1
cookie=0x0, duration=269.257s, table=0, n_packets=263, n_bytes=25546, idle_age=11, in_port=2, dl_src=00:00:00:00:00:02, dl_dst=00:00:00:00:00:03 actions=mod_vlan_vid:4, IN_PORT
```

Was bietet ein  
logisches SDN



# Use Case 1: Geplanter Unterhalt

Operator :  
Switch braucht  
Firmware Update...

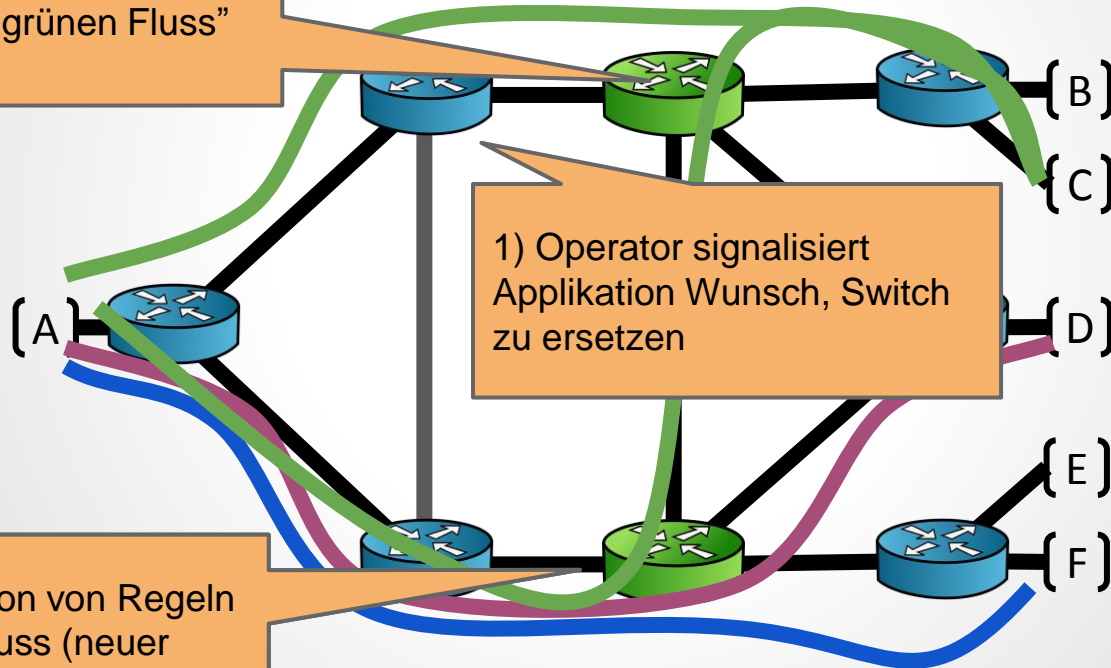


... restliche  
Switches sollen  
kooperieren, um  
Unterbruch zu  
minimieren!

Die **Software** soll sich um Abhängigkeiten  
kümmern, automatisch und korrekt!

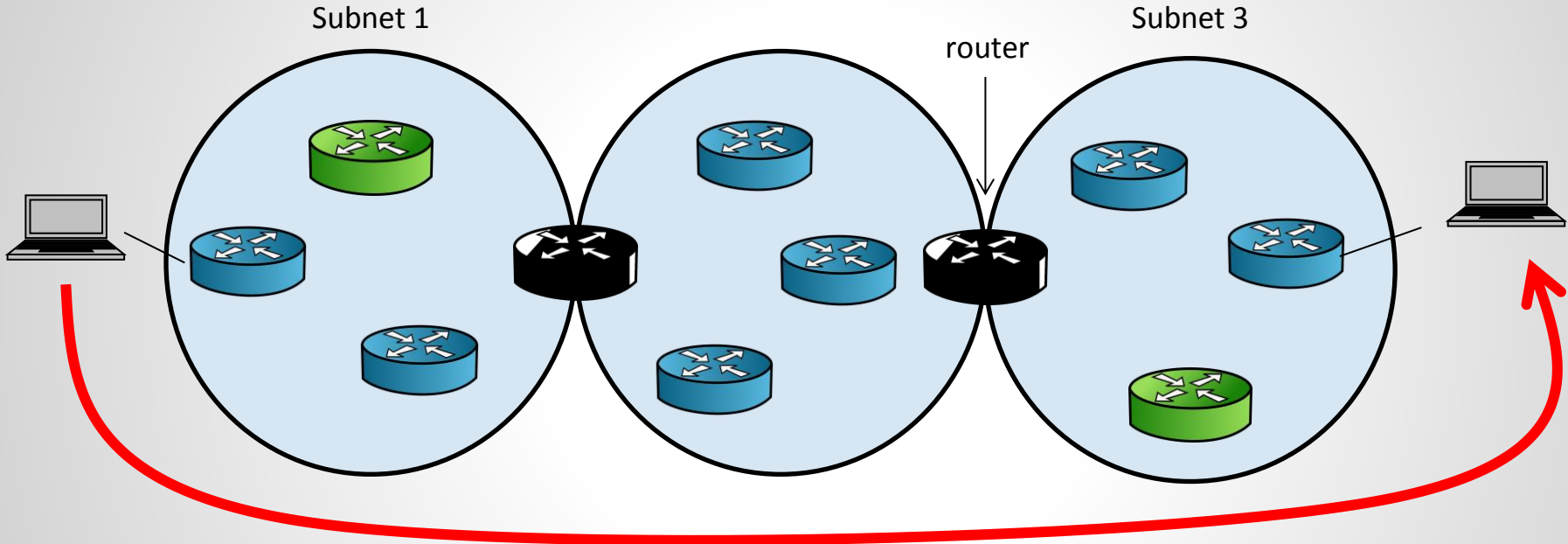
# Use Case 1: Geplanter Unterhalt

3) Update Forwarding Rules:  
ersetze alten "grünen Fluss"



2) Installation von Regeln  
für Ersatzfluss (neuer  
"grüner Fluss")

# Use Case 2: Subnet Mobility



Heute komplex: viele MPLS Tunnels.  
Alternative mit SDN: Logisch zentralisiert.  
Ein SDN Switch pro Mobility Domain reicht.

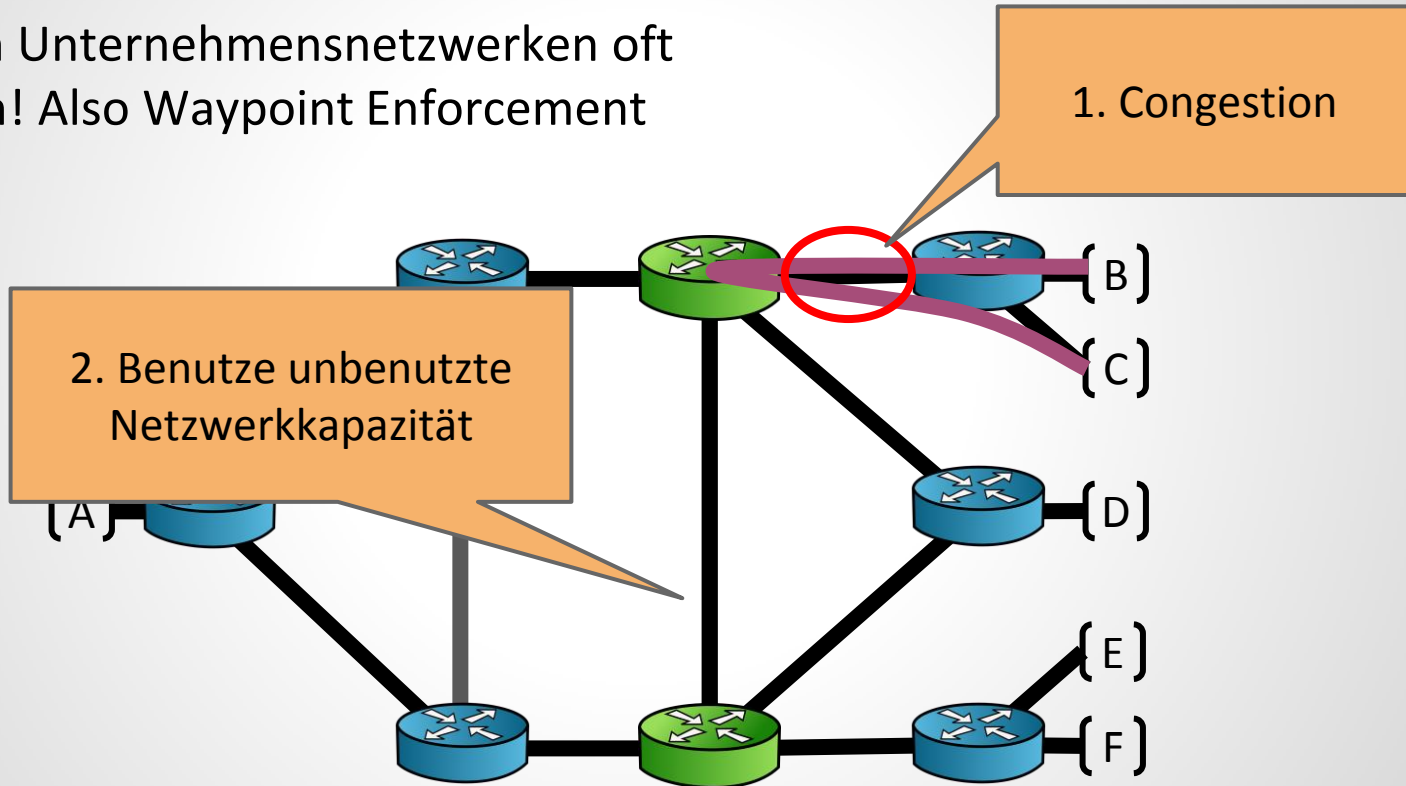
Was ist der Overhead von  
Waypoint Enforcement?





# Auswirkung auf Traffic

Utilization in Unternehmensnetzwerken oft nicht kritisch! Also Waypoint Enforcement "machbar".



# Weitere SDN Anwendungen

## ISPs

- Besseres **TE**
- Ermöglicht **NFV** (in-network services: s. morgen!)

Weitere Vorträge!

## IXPs: Eine natürliche L2 Anwendung

- Ermöglicht flexible **Policy Spezifikation** und **Anwendungs-spezifisches** Forwarding
- Z.B.: Sende **in-bound HTTP Traffic** an **Cache Port**

## Storage Networks

- Trend: Mehr Memory weniger Disk
- **Netz wird zum Bottleneck!**  
Effizienter durch SDN?

## Cellular Networks

- **Fein-granulare Policies** für mobile operators
- **Service Chaining**: basierend auf Subscriber Attributen und Anwendung

## Wifis

- Erhöhe Throughput...
- **Durch Anwendungs-spezifische** Behandlung

## «Billige Router»

- Ersetze Router durch Switches
- FIBIUM Traffic Offloading

Etc.! 😊

**Ein paar Worte zu den Herausforderungen...**

### Challenge 1: Software

Software ist flexibler aber auch unsicherer! Wie kann ich zuverlässige Control Software bauen und verifizieren?

*Don't shoot in your foot!*

der Kontrolle über

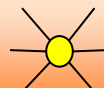
### Challenge 4: *Mind the Gap*

Separation = Verliere Visibility, und möglicher Overhead

### Challenge 5: *Think Local!*

Welche Funktionalität sollte in Data Plane bleiben?  
Wichtig für Effizienz!

Control Programs



Control Programs

### Challenge 3: Logical SDN

Was ist die richtige Abstraktion? Einfach vs ineffizient!

er Platform

### Challenge 2: Sprachen

OpenFlow ist praktisch "Assembler" ("stone-age"): Ich möchte aber high-level Programmiersprachen!

### Challenge 6: Effiziente Regeln

Fein-granular kann teuer sein! Wie kann ich Regeln aggregieren um Memory zu sparen? Achtung: Grobe Regeln sind teuer (in HW) oder langsam (in SW)!

### Challenge 7: Interoperability und Migration

Breche nicht das globale Routing System!  
Lösung: forwarde nur entlang BGP-announced Forwarding Pfaden?

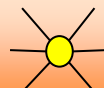
### Challenge 1: Software

Software ist flexibler aber auch unsicherer! Wie kann ich zuverlässige Control Software bauen und verifizieren?

*Don't shoot in your foot!*

der Kontrolle über

Control  
Programs



Control  
Programs

### Challenge 2: Sprachen

OpenFlow ist praktisch "Assembler" ("stone-age"): Ich möchte aber high-level Programmiersprachen!

### Challenge 3: Logical SDN

Was ist die richtige Abstraktion? Einfach vs

er Plattform

# Und vorallem: Wie skaliert man SDNs?

### Challenge 4: Separation und mö

Separation und mö

ich Regeln  
ung: Grobe  
(in SW)!

### Challenge 5: *Think Local!*

Welche Funktionalität sollte in Data Plane bleiben?  
Wichtig für Effizienz!

### Challenge 7: Interoperability und Migration

Breche nicht das globale Routing System!  
Lösung: forwarde nur entlang BGP-announced Forwarding Pfaden?

# Challenges (1): *Mind the Gap!*

- Good News: Separate Control Plane
  - Globale View ist «praktisch»
- Bad News: Separate Control Plane
  - Reduzierte Visibility: Übersehe in-band Ereignisse?
  - Möglicher Overhead / Latenz?

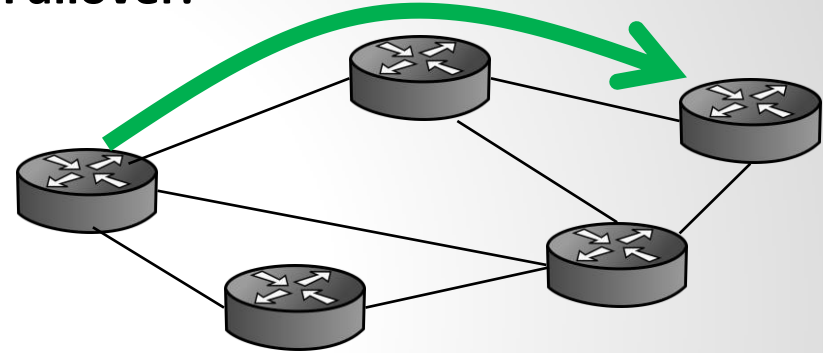
Was kann übersehen werden?  
Welche Funktionalität sollte in der Data Plane bleiben?

Borokhovich et al. (HotSDN'14, HotNets'14): in-band local fast failover Mechanismen für ideale Robustheit, und in-band Funktionen für Troubleshooting

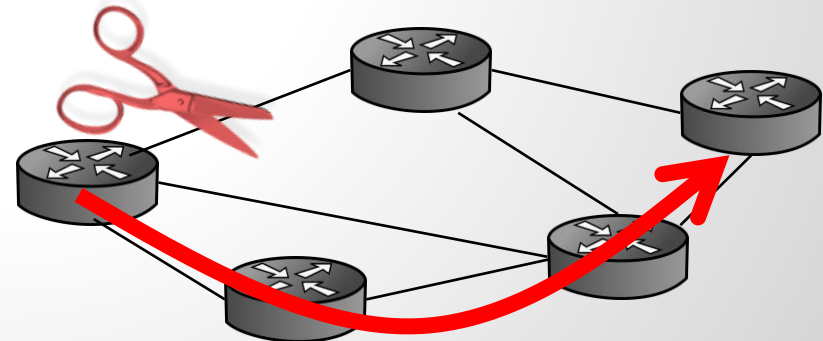
# Challenges (1): Beispiel

- **Link Failures** sind nicht selten
- Moderne Netzwerke bieten **robuste Routing Mechanismen**
  - D.h. Routing reagiert auf Ausfälle
  - Beispiel: MPLS local and global Path Protection

Vor Failover:

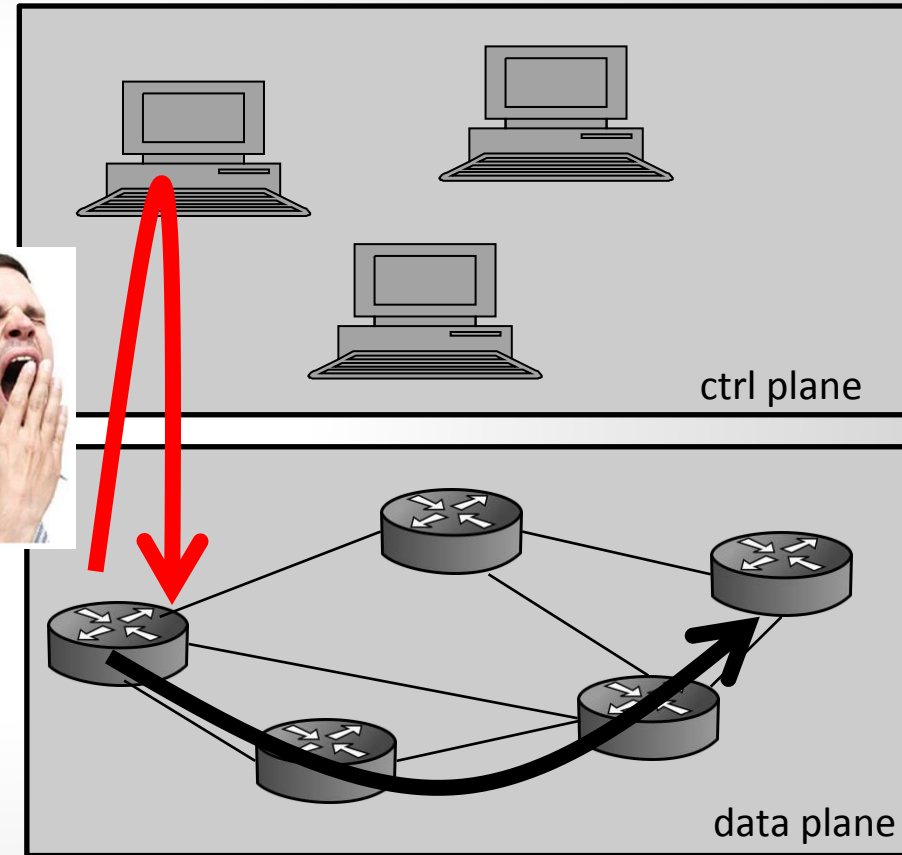


Nach Failover:



# Challenges (1): Beispiel

- Wichtig dass Failover **schnell** = **in-band**
  - Reaktionszeit in Control Plane viel länger als in-band
- Deshalb: **OpenFlow Local Fast Failover Mechanismen**
  - Unterstützen bedingte Forwarding Regeln (abh. von lokalem Zustand des Links: live or not?)
- Bietet schnelles aber lokales und ev. **“suboptimales”** Forwarding
  - Controller kann später verbessern...

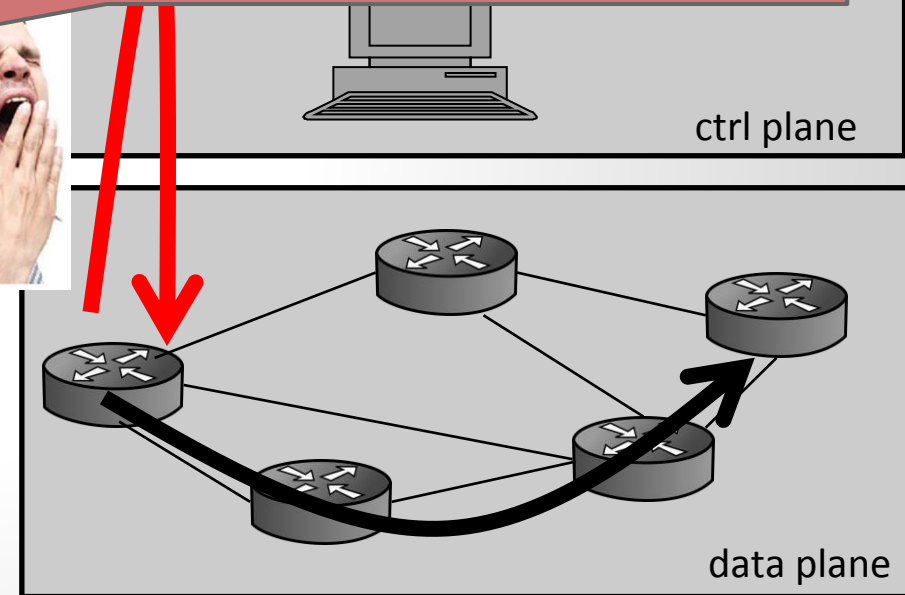




# Challenges (1) Design

Unklar wie man OpenFlow Fast Failover Mechanismus optimal **einsetzt**.  
Z.B.: **Wieviele Failures** können toleriert werden?

- Wichtig dass Failover **out-of-band**
  - Reaktionszeit in Controller länger als in-band
- Deshalb: **OpenFlow Local Fast Failover Mechanismen**
  - Unterstützen bedingte Forwarding Regeln (abh. von lokalem Zustand des Links: live or not?)
- Bietet schnelles aber lokales und ev. **“suboptimales”** Forwarding
  - Controller kann später verbessern...



# Challenges (1) Design

- Wichtig dass Failover **out-of-band**

- Reaktionszeit in Control Plane länger als in-band

- Deshalb: **OpenFlow Local Fast Failover Mechanismen**

- Unterliegen Regeln Link

- Bietet sich ev. "su"

- Con

Unklar wie man OpenFlow Fast Failover Mechanismus optimal **einsetzt**.  
Z.B.: **Wieviele Failures** können toleriert werden?



ctrl plane

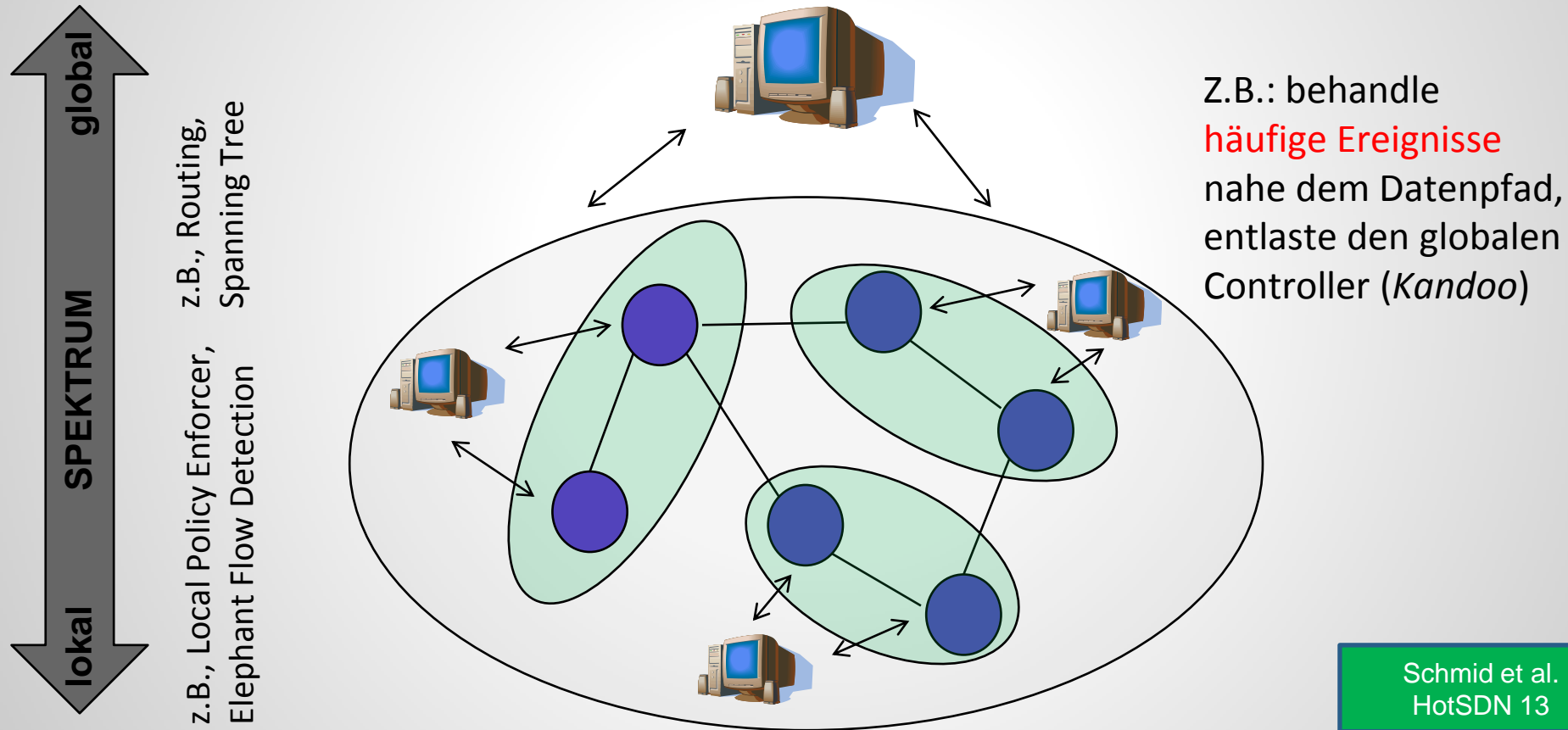
**Nicht-triviales Problem** selbst wenn darunterliegendes Netzwerk connected ist: (1) bedingte Failure Regeln müssen **im voraus** installiert werden, ohne Wissen über tatsächliche Fehler, (2) Sichten zur Laufzeit sind **inherent lokal**.

Wie kann man die Mechanismen nützen **ohne sich in den Fuss zu schiessen** (Loops zu kriegen)?

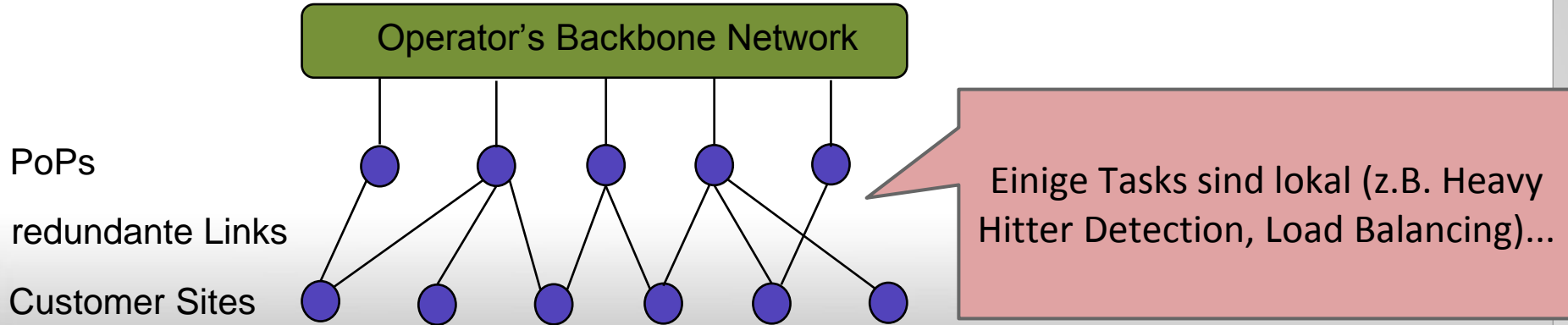
ne

# Challenge (2): Design verteilter Control Plane

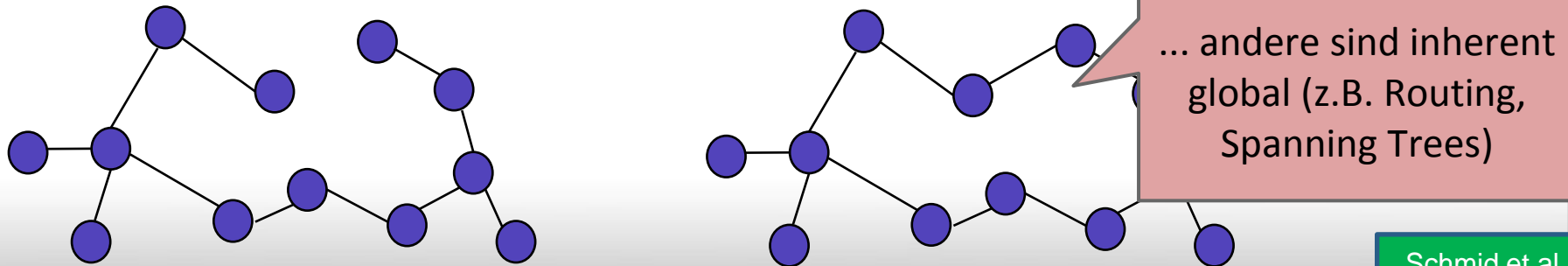
## *Think Global, Act Local!*



## SDN Task 1: Link Assignment („Semi-Matching Problem“)

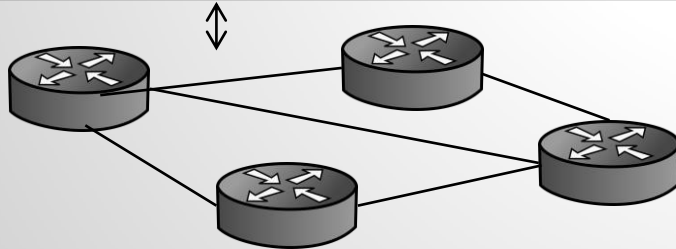
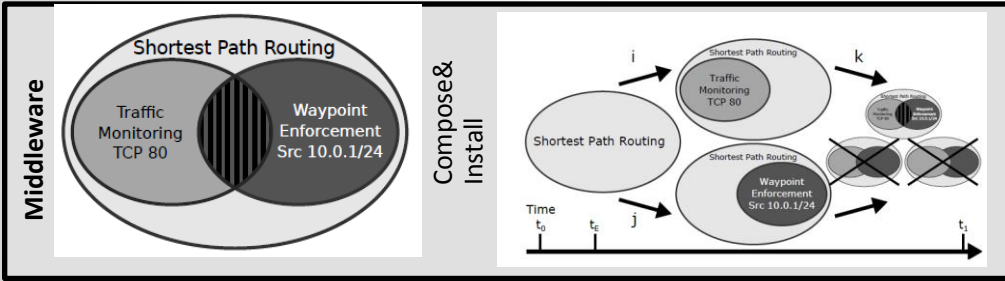
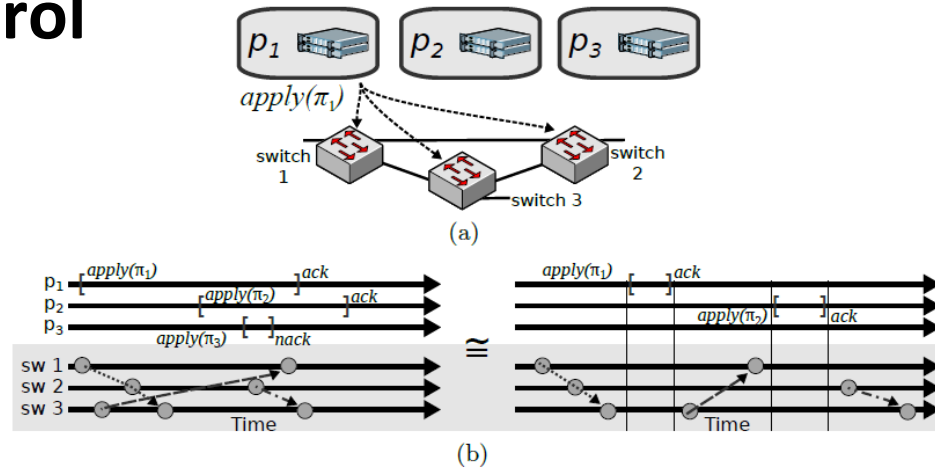
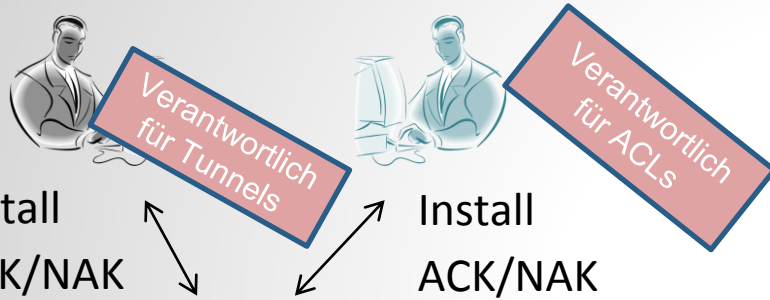


## SDN Task 2: Spanning Tree Verification



# Challenge (3): Concurrent Control

## Compose & Conquer!



**Ziel:** Multi-Autoren Policies

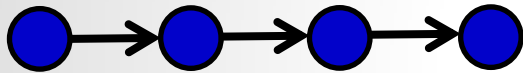
**Problem:** Konflikt-freie, per-packet konsistente Policy Komposition und Installation

**Heiliger Gral:** Linearizability (*Safety*), Wait-freedom (*Liveness*)

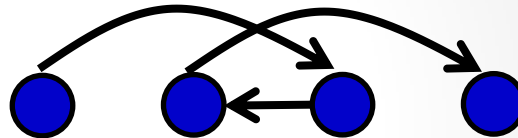
# Challenge (4): Konsistente Policy Updates

*Exploit flexibilities but: Zentral ist nicht trivial!*

Alte Policy

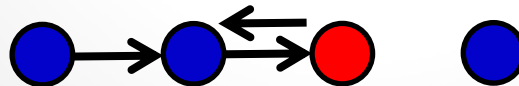


Neue Policy



Nur schon Update von  
single Switch ist nicht  
trivial! Mehrere erst  
recht nicht...

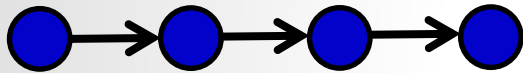
Inkonsistentes Update



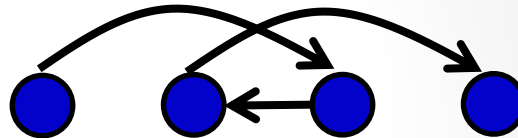
# Challenge (4): Konsistente Policy Updates

*Exploit flexibilities but: Zentral ist nicht trivial!*

Alte Policy

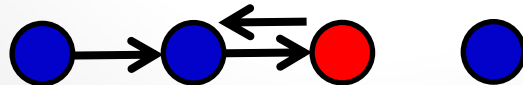


Neue Policy



Nur schon Update von  
single Switch ist nicht  
trivial! Mehrere erst  
recht nicht...

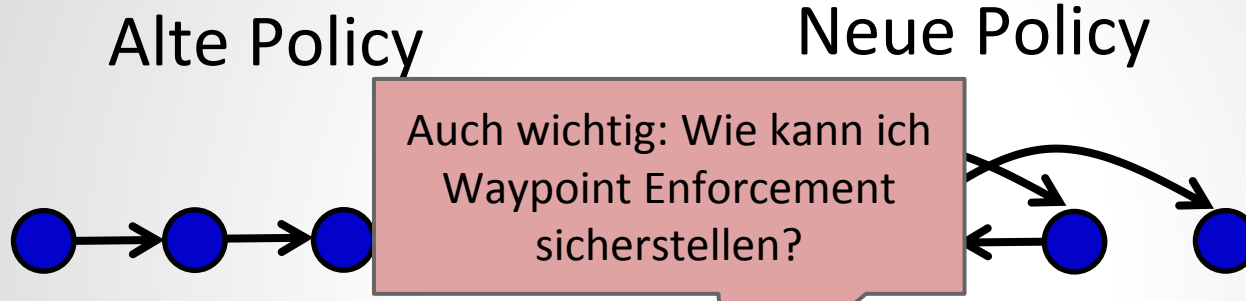
Inkonsistentes Update



Ein Tradeoff zw. Update  
Speed und Konsistenz?

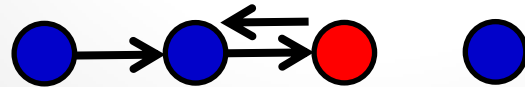
# Challenge (4): Konsistente Policy Updates

*Exploit flexibilities but: Zentral ist nicht trivial!*



Nur schon Update von single Switch ist nicht trivial! Mehrere erst recht nicht...

Inkonsistentes Update



Ein Tradeoff zw. Update Speed und Konsistenz?



# SDN Challenges: Und viele mehr...

- **Participatory** Networking / Computing: Wer sollte welche Kontrolle haben? In der Vergangenheit haben Forscher eher ums Netzwerk herum gearbeitet als mit dem Netzwerk...
- Wie können **SDN Domänen** mit non-SDN Domänen interagieren?
- Etc.

# Zusammenfassung

- Netzwerke werden **virtueller, software-definiert**, und **offen**
- Grosses Interesse: Facebook, Microsoft, Deutsche Telekom **unterstützen Open Networking Foundation** und fordern offene Standards
- Inkrementelles Deployment möglich (z.B. für **Confidence Building** oder um **Kosten** zu sparen)
- Fundamentale **neue Herausforderungen** für Forschung und Industrie (z.B. verteilte Kontrolle und Updates)
- Killer **Use Case**? Die Zukunft wird's zeigen: SDN hat erst begonnen...

# Danke!

## Literatur:

- Canini et al. (**USENIX ATC 2014**): Migration zu SDN mit Panopticon
- Borokhovich (**SIGCOMM HotSDN 2014, ACM HotNets 2014**): Robuste in-band failover Mechanismen und andere OpenFlow Funktionalität
- Canini et al. (**SIGCOMM HotSDN 2013**): Parallele Control Plane
- Schmid et al. (**SIGCOMM HotSDN 2013**): Verteilte Control Plane
- Ludwig et al. (**ACM HotNets 2014**): Konsistente Netzwerk Updates mit Waypoint Enforcement
- Feamster et al.: The road to SDN

Wir bieten hands-on SDN Tutorials an. Interessiert? Für mehr infos: [stefan.schmid@tu-berlin.de](mailto:stefan.schmid@tu-berlin.de)

# Hands-on Tutorials

← → ↻ <https://trial6.badpacket.in/topology#> 🔑 ☆

exercise\_1 ⚙️ No Changes

👁 Physical Logical

```
graph TD
    A[A] --- sw5[sw5]
    sw5 --- sw3[sw3]
    sw3 --- sw1[sw1]
    sw1 --- sw2[sw2]
    sw2 --- sw4[sw4]
    sw4 --- sw8[sw8]
    sw8 --- F[F]
    sw4 --- sw7[sw7]
    sw7 --- D[D]
    sw2 --- sw6[sw6]
    sw6 --- B[B]
    sw6 --- C[C]
```

4. Note whether any changes have appeared in the flow table of **sw2**.

5. Look again at the MAC table of switch **sw6**; observe the VLANs used by traffic traversing this switch

### Challenge

What is the designated SDN switch for the source-destination pair B and C?

### Hints

To view a flow table, select an SDN switch in the network topology and click on **Show Flow Table** in the bottom action bar.

Look at the logical network view.

! Take-Aways

All traffic between B and C must traverse an SDN switch because B and C are connected to the network via SDN controlled ports.

You are now ready to move to the next task.

sw2

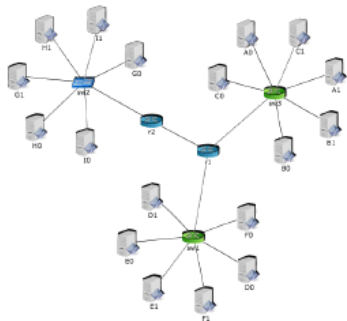
Every 1.0s: ovs-ofctl dump-fl... Fri Sep 19 18:25:37 2014

```
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=269.257s, table=0, n_packets=263, n_bytes=25546, idle_age=11, in_port=2, dl_src=00:00:00:00:00:02, dl_dst=00:00:00:00:00:00:03 actions=mod_vlan_vid:4, IN_PORT
 cookie=0x0, duration=388.481s, table=0, n_packets=9, n_bytes=890, idle_age=380, in_port=4, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:00:04 actions=mod_vlan_vid:7, output:1
 cookie=0x0, duration=269.257s, table=0, n_packets=263, n_bytes=25546, idle_age=11, in_port=2, dl_src=00:00:00:00:00:02, dl_dst=00:00:00:00:00:00:03 actions=mod_vlan_vid:4, IN_PORT
```

<https://venture.badpacket.in/training/>

# Hands-on Tutorials

## Hybrid SDN Deployment & Operations



## hands-on tutorial & training

Interactive tutorials and hands-on training approaches for deploying and operating software defined networks

## Contents

A 3-hour training course covering approaches for deploying and operating Software-Defined Networks in existing network deployments.

The training covers:

1. Introduction to core SDN concepts
2. Hybrid deployment and operation of SDN illustrated via hands-on exercises
3. Enterprise subnet mobility use case

## Audience

The training targets network operators and professionals, who want to gain:

1. a stronger understanding of SDN applications and their benefits
2. how to introduce and realize them incrementally in existing networks.
3. hands-on experience configuring legacy enterprise network infrastructure to interact with SDN devices

## About us

We are a Berlin based team of experienced network researches and engineers with a strong track record of innovative networking management, testing, and control solutions.

Please contact us for a training programme tailored to your unique needs. Remote and on-site offerings available.

Mail: [info@badpacket.in](mailto:info@badpacket.in)

Phone: +49 30 314 78753

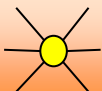
Web: <http://venture.badpacket.in>



# Backup Slides

# SDN Interface

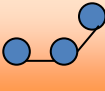
Control Programs



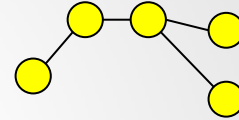
Control Programs



Control Programs



“Logically Centralized”  
Global Network View



Controller Platform

OpenFlow (Data Plane API)

